AMIGA COMPUTING

Q&A

Question after question answered by Amiga experts

Printers • Disk drives
AmigaDOS • Graphics
Sound and lots more!

Europress Enterprise Ltd
Europa House
Adlington Park
Macclesfield
SK10 4NP
Telephone: 0625 878888
Fax: 0625 850652

Amiga Computing is the complete solution for all Commodore Amiga owners. Out from Europress Enterprise on the first Thursday of each month, it's packed with authoritative features, guides, reviews, previews, news and games – plus the best cover disk in the business. Don't miss out!

First published in the UK by Europress Enterprise Ltd, 1993

Written by Paul Overaa, Mark Smiddy, Phil South and Gary Whitley
Production by Eddie Burke and Phil Morse

ISBN 1 897628 03 X

Printed and bound in Great Britain by Cox and Wyman Ltd, Reading

CONTENTS

Q&A

AMIGADOS

AmigaDOS is without doubt the most complex and most comprehensive device operating system on any micro computer with the exception of Unix and that requires at least a 68030 to run. Perhaps it should not be too surprising that many people find it so confusing at first. The best way to learn about AmigaDOS is by hands-on experience – no book in existence can teach you this, but a good reference work would not go amiss.

AmigaDOS comes in many flavours. Four are in common use: 1.2, 1.3.x, 2.x and 3.x. 1.2 and 1.3 for 1.2 or 1.3 Kickstarts; 2.x for 2.0 Kickstart and 3.0 for 3.0 Kickstart. Broadly speaking the version of AmigaDOS you are using depends on the Workbench you have.

AmigaDOS has never been a subject which lends itself to simple Q & A sessions – everyone has their own particular problem. The questions answered here are broadly based on the vast range of queries the author receives daily. The technical level ranges from beginner to quite advanced so do not worry if some answers seem well above you; you can always come back to them as your knowledge grows. In all cases a basic knowledge of how to open a Shell and enter commands is assumed.

Q: Is it possible to use CD with a pattern?

A: Yes – provided you have AmigaDOS 1.3 or higher. The basic syntax for all systems is:

```
SPAT CD
```

an example to get to the keymaps directory could read:

```
SPAT CD :DEVS/K#?
```

SPAT is a script which offers simple pattern matching to commands which do not already have it. DPAT by comparison does much the same for commands which have two required arguments: RENAME for instance. Care should be taken using this command since it will not warn you if more than one possible path matches the pattern.

In Workbench 2.0 and higher, CD itself has pattern matching, and therefore does not need SPAT or suffer from the problems (if the pattern matches more than one path, you will be warned and it will take no action). More

interestingly though, it is possible to use more than one pattern in the search, viz.:

```
CD :D#?/K#?
```

Also, if you don't mind spelling out the complete directory pathname, there is no need to enter CD. The following examples are all valid:

```
DEVS:   ; Change directory to DEVS.
System  ; Enter the system directory.
:   ; Go straight to the root of this disk.
//  ; Move two levels up the hierarchy.
```

Q: DiskDoctor says I have a "hard error" on my disk – is it ruined?

A: DiskDoctor is a much berated program which has been removed from Workbench 3 "for malpractice" as one developer put it. However, once a disk develops a fault the data is lost. Typically these faults occur on the root directory at key (block) 880 – mainly because that particular block gets the most hammer. If this happens you will be unable to access any data on the disk because every single file and directory on the disk is attached to it – however remotely.

DiskDoctor can often rescue things in this case and allow you to get most of your data back. The program is often blamed for dumping files all over the disk, but in truth it is just making the best of a bad job. In any instance, once you have run DiskDoctor, you must copy the files from the bad disk and reformat it – 90 per cent of the time a clean format will cure the fault. Interestingly, it can even rescue a disk that has been formatted with the Quick option.

Q: What is a path?

A: The best way to answer this is to think of what a path is. What do you use paths for? To get to places! This is precisely what AmigaDOS does with paths; so this historical term could have been roads instead. A path tells AmigaDOS where to find something – a route around the directory hierarchy if you like.

When you enter a command (the first word on the command line) AmigaDOS looks down each path and checks if the command is there. It starts with the current directory and continues down the path list until it either finds the command or runs out of paths and generates an error. Internally AmigaDOS looks up its commands using a method called "hashing" and can search even a large path list in fractions of a second.

While this method is highly efficient, it can give rise to some odd problems if you are not careful where you put things. Say for example you put a file

called "DIR" in the RAM disk and then try to execute the command AmigaDOS will act like this:

```
DIR
dir: file is not executable
```

In case of problems like this you can use the WHICH command to discover where the problematical command is:

```
WHICH DIR
Ram Disk:dir
```

Q: When I use EVAL it sometimes gives odd results, why?

A: EVAL – the expression evaluator – is bugged in early (1.3) releases of AmigaDOS and only uses the standard command line parsing to retrieve its arguments. This can be avoided by making sure you always separate the argument by at least one space:

```
EVAL 1+1     ; this is wrong
EVAL 1 + 1   ; this is OK
```

Alternatively, you can use the keywords in the parser, viz.:

```
EVAL value1=1 op=+ value2=1
```

such a technique is very useful in a script file.

Q: What is a volume?

A: A volume is just another name for a disk. The fastest way to reference a disk is to pop it in any drive and specify the drive number. For instance, you might enter something like this:

```
DIR DF0:
```

which would attempt to list the contents of the disk in the internal drive. This technique is anything but perfect though, as many first-time single-disk users soon discover. Remove the Workbench disk, pop another disk in the drive, enter the command and AmigaDOS asks for the Workbench disk back, at which point it lists the Workbench disk's contents.

This is because AmigaDOS is looking for the DIR – located on the Workbench disk. Once loaded, the command executes immediately and lists the contents of the disk in drive, DF0: – precisely what you asked it to do.

Irrefutable proof that computers are stupid!

It is possible to avoid this by entering the name of the volume to be listed, for instance to list a disk called Extras:

```
DIR Extras:
```

Note the use of a colon (:) after the disk's name – this signifies a volume name.

Q: Using a volume name seems a long-winded way of doing things – is there a better way?

A: Volume names are essential in some circumstances and must be used in scripts, for instance, to make sure AmigaDOS gets the right disk. However, operations such as DIR can be simplified: not least by getting an extra drive or two. The exact technique you have to use depends on the version of Workbench you have. The oldest method, used for Workbench 1.0 through to 1.2 is as follows:

1. Enter the command followed by a query and press Enter, for instance:

```
DIR DF0: ?
```

2. Replace the Workbench disk as requested and wait for the command to load. After loading, it will present a "template" and pause.

3. Place the disk you were interested in in the internal drive and press Return once more.

That solution works – but it is hardly satisfactory because it relies on pre-loading the command. Once DIR is finished, its code is "unloaded" (removed) from memory thus freeing resources. A much better solution was implemented in release 1.3 as most AmigaDOS commands were made pure. This means they could be permanently loaded in memory and always available. The resident command does this, for instance:

```
RESIDENT C:DIR
```

Note that you must supply a complete path for the program to load and it must be pure (most AmigaDOS commands can assumed to be pure; if not, the resident command will warn you and refuse to load them. You can check if a command is pure by listing it and looking for the "P" flag in the protection bits).

Q: What does pure mean?

A: Pure code is not an exclusive preserve of AmigaDOS – it is found in all multitasking environments. It signifies the software is both re-executable and re-entrant. These terms are a little vague and while an in-depth explanation requires more space than allowed, some description is warranted.

When a program is made resident in memory, its code can be accessed many times and by several processes at the same time. This means it cannot contain any internal variable space because those variables must be fresh every time the program is called.

These problems can be avoided by using "local" data areas which are created and initialised every time the program code is executed and lost when that invocation is finished.

Q: When I attempt to sort a large file, the computer crashes – why is this?

A: The AmigaDOS SORT command is based on a recursive algorithm (Quicksort) which continually calls itself. Every time the program calls itself, several bytes of "process stack" are consumed. The depth of recursion, and therefore the amount of stack used, varies considerably according to how much data is being sorted.

Since Exec (which manages the stacks) does not have any stack checking it is possible for data to fall off the end and wrap around on itself – thus destroying the stack integrity and causing a guru.

The simple answer is to increase the amount of process stack temporarily like this:

```
STACK 16000
SORT MyFile
STACK 4000
```

At the end of the sort the amount of stack space is reset back to the default setting of 4,000 bytes. If this technique is to be used inside a script, further error checking is required just in case there is not enough memory to raise the stack. This is accomplished thus:

```
FAILAT 11
STACK 16000
IF ERROR
    ECHO "Sorry, not enough memory available..."
ELSE
    SORT MyFile
ENDIF
STACK 4000
FAILAT 5
```

This is based on the theory that if a large enough stack cannot be allocated, the STACK command returns an error flag (level 10). This would normally stop a script so the failure level must be raised above this – to 11 in this case. After the stack command has been executed, the error condition is tested and appropriate action taken. Whatever happens, at the end of the conditional test the default stack and failure levels must be reset back to default.

Q: What is multitasking and how does it affect AmigaDOS?

A: Multitasking is the ability to run more than one program (apparently) at the same time. The crucial thing here is apparently – each program runs in what can be termed a virtual computer. Insofar as any task is concerned it is the only program running and has sole access to all the resources (memory, hardware, the processor and so on). The operating system looks after all this by invisibly sharing processor time between all the tasks – at any one moment therefore, one program is running and all the others are waiting for something to happen.

Resources are shared by clever programming: a task cannot assume something is available unless it checks first. You can liken this to a program written for many variations of the same computer. Some models may have 8Mb RAM, printer and serial ports plus a hi-res display. Others may have 1Mb RAM, no printer or serial port and a low-resolution display. When a task starts, it can check for the machine's configuration and make the appropriate adjustments.

Therefore, if a task starts and grabs the printer port for its own use, other tasks starting later will find a machine configured without a printer. Tasks, therefore, cannot afford to make assumptions about the environment they exist in. In fact, if something gives up the printer port (for instance) another task can then start to use it as if the port had just been fitted. All this might seem complex, but in the event, as a user you will rarely have to worry about it.

This brings me to an interesting point regarding tasks and processes: the two terms are often confused but are not the same. Tasks are operated by Exec (a major chunk of Kickstart) and operate at very low level. Processes are part of AmigaDOS and ride on the back of tasks. Every new process you start creates its own task – and that is all you need to know: the whole process is automatic.

Q: The MS-DOS system I have used has a command called MORE, with the syntax:

```
TYPE filename | more
```

– does AmigaDOS have an equivalent system?

A: Yes and no – the closest AmigaDOS equivalent to this command in AmigaDOS is:

```
MORE filename
```

and although the result is the same it is achieved in a different fashion. The MS-DOS command uses an unnamed "pipe" which directly sends the output from TYPE to the input of MORE. The MS-DOS command cannot be executed directly since it only works with pipes. Although the system is admittedly powerful, it has a very limited number of uses and the example given is probably the only one in common use.

AmigaDOS pipes are much more accessible and far more useful in practice since they can be used to connect two processes together. A more useful version of the AmigaDOS command is to multitask MORE like this:

```
RUN MORE filename
```

In this example, MORE opens its own interactive window. This has the advantage of not deleting the original Shell window's contents. You can also omit the filename in this case and MORE will prompt you for a file to view. This works best in the later versions where the command will present a file requester.

Q: My Workbench disk has suddenly started taking a long time to load – why is this?

A: This is not an uncommon fault and is caused by the disk validator – a separate process which ensures every disk you put in the machine is free from faults. Certain errors are not serious and can be corrected automatically and one of these typically manifests like this. The simple answer is to write-enable the disk, pop it in the machine, let the validator fix things and write-protect it again.

So how does this happen in the first place? Well, when AmigaDOS finishes writing a file, the validator makes some changes – a short write occurring about half a second or so after completion. If you reset the machine before it gets chance then write protect the disk, the validator will get confused and effectively slow down the boot sequence. Typically, this sort of error is only caused by folk being a little over-enthusiastic while editing the Startup-sequence.

Q: I've seen the expression "> NIL:" appear all over the place – what does it do?

A: This is two things in one and is telling AmigaDOS to send all OUTPUT

from a command to the NIL: device. The arrow suggests direction – > i
outward in this case; < would be inward or INPUT.

NIL: is a "dummy" device maintained in software which absorbs whateve
printing a command generates like a sponge. It could be called to th
DUSTBIN: device because that's exactly what it does.

Q: Why doesn't RAD work on my vanilla A500+?

A: Someone dropped a clanger at Commodore when the distribution disk
were being set up and RAD's mountlist entry was given a HiCyl value of 7
This puts the size of RAD at 880k – almost 1Mb which is far too much for
basic machine. You should reduce the HiCyl value to something more sensibl
say 19 which will give you a 220k reset-proof RAM disk.

Q: If the Amiga uses 1Mb disks why is there only 880k available?

A: The actual amount of data you can store on any particular disk is ver
dependant on the type of data being held. A pure data disk with nothing on
except one large file would have a storage capacity of around 83k (corrected
AmigaDOS 3's INFO command). You lose 120k of possible data spac
because the disk is divided into 1,760 little blocks or sectors and the ext
capacity is lost between the gaps.

The other 43k is lost due to the way information is stored using OFS. Eac
of the 1,756 available data blocks can only hold 488 bytes of your data; the re
is pinched by the system for its own use. In the real world, blocks are used b
things like directories and file headers and any disk's capacity is very difficu
to quantify exactly. In fact, a disk could be filled by putting 1,755 us
directories on it, if you had the patience to do it!

Q: Why is the RAM disk always 100 per cent full?

A: It just works that way. The RAM disk is a disk simulation which has a si
defined by its current contents. RAM: grows and shrinks according
requirements but it can never be larger than a floppy disk.

JARGON BUSTER

data redundancy – A measure of how well a storage device can recover fro
a media failure. The higher the redundancy, the better the chance. Measured
a percentage, 100 per cent redundancy can only be achieved by simultaneo
backup to a secondary device.

device – Classically a device is some hardware that does a job. The nam

"device" in AmigaDOS refers to the pseudonym (one or more letters followed by a colon) attached to the device driver – the software that communicates with the hardware (electronics). There are two types of device in AmigaDOS, real devices such as the printer, disk drive or serial port and logical devices like pipes, RAM drives and consoles. As far as AmigaDOS is concerned, there is no difference. At AmigaDOS level, all devices support one or both of the following modes: INPUT – data is read from the device; OUTPUT – data is written to the device.

FFS – Fast Filing System. A development of OFS which is faster and has a slightly improved storage potential at the expense of a reduction in data redundancy. FFS is mainly used on hard disks but can be used on floppies from release 2.

OFS – Old Filing System. The original filing system used on hard and floppy disks. Although slower than FFS, it offers much better data redundancy – which is why it is the preferred choice for floppy disks.

Vanilla – Plain, un-expanded, basic hardware. Also a common flavour of ice cream.

GRAPHICS

There are a lot of graphics applications out there, and a lot of peripherals which make life worth living if graphics is what you want to do with your AmigA: What's the best program and system configuration to use? Where do you get it? How much does it cost? Graphics on the Amiga is a tricky business, and if you've bought your Amiga with a view to doing graphics it can be a little bit daunting having all this seemingly contradictory information flying about: AA chipset is best, 24-bit is best, two-colour lo-res with a mouse taped to your nose is best... that kind of thing. It's all very confusing.

Let's calm your nerves with a quick question and answer session about all your commonly-hit problem areas and furnish a few full-on answers.

Q: What sort of graphics can I do on the Amiga?

A: Graphics on the Amiga fall into a number of categories. The graphics you commonly see on the screen are bitmap graphics – dots of colour on the screen. Another kind of graphics are meant to be printed out or used in publishing. These are called structured graphics. Graphics for video have to be very high resolution, and these are usually 24-bit graphics, although for reasons we'll go into later the new Amiga 1200 and 4000 come close to the right quality without

the need for true 24-bit.

Also you can use programs to treat images you have already created, and there are also morphing programs which let you make animations blending one image to another (the advert for BBC TV with Smith and Jones is a good example of this, also films like James Cameron's Terminator 2).

Bitmapped graphics are simple to explain. Each coloured dot on the screen is defined by a series of "bits", or digital locations inside its memory. These bits can store the location of a point on the screen, what colour it is, and what shade. The bits form a "map" or pattern across the screen, telling the computer where each dot is and what colour it should be. The creation of bitmap graphics is usually called painting.

Structured graphics are used in graphics programs destined for use in publishing like Professional Draw 3 (Gold Disk). This creates graphics which are jaggy and horrible on the screen, but are nice and smooth on the page when printer out. This is because the graphics are created on the printer rather than in the computer for the screen, and so are tied to the resolution of your printer rather than your Amiga.

Morphing is easy once you have the right programs. Morph Plus (ASDG) specialises in the creation of morphs, metamorphosing one picture or animation into another. Basically what you do is load one sequence or still frame, load another, and show on the pictures which bits of the first you want to relate to the second.

Fractals are mathematical shapes which have the remarkable ability to imitate real life objects with amazing accuracy. Many of the fractal programs available mimic trees, plants, mountain ranges and other natural phenomenA: Fractal pictures grow like time lapse pictures before your eyes, and all you do is set up the initial position for the creation of the picture to happen automatically.

One of the best programs for this is Vista Pro (Virtual Reality Labs) which is a 3D fractal landscape maker. All you do is tell the Amiga where the light is coming from, load a file of a landscape (some of those on disk are digitised from real landscapes) and click the mouse button. All the contours of the landscape are simulated with fractal curves, making the digital map into a realistic landscape before your eyes.

Q: What are 24-bit graphics? Do I need extra hardware?

A: Twenty-four bit graphics has become something of a buzz-phrase in the Amiga community, nay in all computer fields. Most graphics you see on Amiga are five-bit, sometimes six or 12. The bits refer to the amount of "bitplanes" the graphics use, and the more bitplanes the more colours at the rate of two colours per bitplane. So using that as a basis, one-bit images are two colour, two-bit images four colour, three-bit have eight colours, four-bit

have 16 colours, five-bit have 32 colours, six-bit have 64 colours, 12-bit (HAM) images have 4,096 colours, and 24-bit have 16.8 million colours.

Just because you have this amount of colours available doesn't mean you can have all the colours on screen at once, as there aren't 16 million dots on a monitor screen. What it rather means is that you can choose your colours from a palette of 16 million.

Although you can simulate 24-bit graphics on an AA chipset Amiga like the 1200 or 4000, you need extra hardware to get true 24-bit graphics out of your Amiga. In recent years the price of 24 bit graphics has come down radically, from about £2,000 to about £200–£600.

A more low end solution is the DCTV (Digital Creations). The 256-colour DCTV has a built-in frame grabber and paint program. Although users of the 1200 and 4000 will have no need for the DCTV, as the AA chipset does much the same thing, other Amiga users will be very happy with the results, giving entry to the loftier heights of Amiga graphics for very little cost.

The Harlequin card (Amiga Centre Scotland) was one of the first display boards to handle 24-bit. Although quite expensive, it does deliver very high quality and is first choice for professional users.

OpalVision (Opal Technology) is one of the most recent innovations in 24-bit technology – a board that fits into an Amiga video slot which displays 24-bit graphics on a normal Amiga monitor without the need for a separate display. The board is soon to be put into a unit for use with the A500 and even the A600/1200.

As well as 24-bit graphics and the excellent OpalPaint paint program, the OpalVision is soon to include Digital Video Effects, and a frame grabber and genlock, to make it the perfect Amiga video solution.

Q: What ways can I get output on paper from my Amiga?

A: You need a printer, and almost any will do, but which you choose depends on what kind of output you want. If you want colour output then the cheapest option is a colour dot matrix like the Citizen 224, and the best option is a colour inkjet like the Hewlett Packard Deskjet 500C. The dot matrix is smudgier but cheaper, the DeskJet is sharper and clearer but more expensive to run with its ink and special paper if you want top quality output.

If it's black and white you want almost all the dot matrix jobs are not worth the bother, especially when you can get a black and white inkjet for about £300.

Either the Canon BJ20 or the HP DeskJet 500 will fit the bill. At the top end, and if your needs are more in the DTP line, you'll have to lash out on £700 worth of laser printer, and either the Ricoh LP1200 or the Epson ELP4300 are best for that.

Q: Can I put Amiga graphics on anything other than paper, a T-shirt for example?

A: One of two ways – do it yourself or get a company to do it for you. To do it yourself you need to print out your design on paper, and either make a stencil yourself or get one made. Your local silk screen supplier can fit you out with the frames, silk, photo-sensitive materials and inks to get started. You have to print the design out, colour separated, on sheets of paper and have that image photographically transferred onto a screen, one for each colour of the design. Then using a rubber squeegee you press ink through the screen and onto the shirt. Many local colleges run courses on screen printing, which incidentally also gives you access to high class equipment for a few weeks!

Obviously you can send your design away to be printed professionally. Some firms can do it direct from Amiga disk like Silicon Picture Company 081-556 7607. But you can have your colour separations put on paper and give them to a conventional printer.

Q: I can't draw. Will a computer program help me?

A: To a certain extent, yes. Computers are good at amplifying what you do and making tasks which are difficult by hand easier. This idea can be extended to graphics. If you can't draw a straight line or a curve then a graphics program will help you a lot. If you got a lot of ideas but lack the expertise in putting them down on paper then a Amiga is your man. What the Amiga can't give you is ideas.

Q: When I use my graphics package, I keep getting an out of memory error. Why does this happen and is there anything I can do about it?

A: As you'll see from some of the answers given here, the word on the street is "more", particularly more memory. This means further outlay but is it worthwhile? Well, let's start by making this clear – graphics is a very memory-intensive process, and if you only have 1Mb of memory the amount of real graphics you can do is going to be a lot but definitely finite. You will hit problems as you move up in resolution and colours, to the degree that HAM mode animation is not really possible in a 1Mb machine – there just isn't the headroom.

So the amount you spend on extra hardware should be balanced against what you intend using the machine for. Are you a professional? Or are you a serious recreational user? How serious? Do you just want to play with graphics, or are you after making a serious artistic statement? All these questions will need answering if you're going to make any kind of decision. Expanding your fast RAM using trapdoor expanders or memory cards is all well and good, but the only way to banish the problem for ever is either to have the full 8Mb or RAM

or start add chip memory. 2Mb chip upgrades are available for all Amigas.

Q: What does "digitising" mean?
A: Digitisers are bits of hardware which take an image from a video signal and turn them into IFF bitmap graphics. Some like VIDI 12 (Rombo) do great colour grabs from colour video signals up to HAM resolution. Others like the VLab (MacroSystem) grab YUV format (easily convertible to 24-bit) pictures from video in real time.

Q: Can I do 3D graphics on my Amiga? What are the best programs to use?
A: 3D graphics are made by rendering rather than drawing by hand, a bit like fractals. The word rendering means graphics created by calculation rather than hand drawn. You set up the basic circumstances of the rendered picture so ir a sense even though the design of the shapes etc is yours, you're made to feel slightly less responsible for the process than you would be if you were actually drawing it.

3D graphics are created by first modelling the objects and lighting, which are then processed by the program and displayed on the screen in any screen resolution from 16 colour hi-res or HAM all the way up to 24-bit.

Imagine 2 (Impulse) does all this and more and lets you make the kind of 3D graphics you see on TV, provided you have some kind of 24-bit display option, that is. The objects can be created in the program, saving a series of pictures to disk which can then be either combined on videotape a frame at a time using a single frame controller, or saved as a HAM animation.

Another option is Real 3D (Activa). Although it's not as professional as Imagine, it still has a lot to offer. The beginners version of this program is perfect for the person starting out in 3D.

Pixel 3D (Axiom Software) is essential if you're serious about 3D as it enables you to convert almost any object from another 3D program into any other format. Soon to be available in an extended version as Pixel 3D Professional.

As far as system configuration goes, you'll need maximum memory and possibly even an accelerator to get anything much out of these programs. So 3D is not for the poor or tight of pocket.

Q: I have some PC and Apple Macintosh graphics which I want to use on my Amiga system. How do I transfer them, and can they be converted in any way? If so can you do this the other way too, transferring Amiga graphics to PC and Apple Mac?
A: Once you've transferred the graphics to an Amiga disk using CrossDOS

(Consultron) or A-Max II (ReadySoft) you can convert the images using one of a variety of image processing programs. These can convert pictures from other computers, or from different IFF resolutions to the one you want. Most also feature some very nice image processing algorithms, like turning them into 3D embossed on stone, re-colouring them and scaling them up and down in size.

The first you should consider is Art Department Professional v2.0 (ASDG) which features the ability to convert almost anything to anything else, and drivers for scanners, 24-bit cards and transport controllers. Needs lots of memory to do its job, but then anything with the word "professional" in the name will!

The only other option is Image Master (Black Belt Systems) – a great program with a lot of very eccentric options, but again very memory hungry, and an accelerator card wouldn't go amiss either! Converts formats, does image processing, even does morphing and is upgraded almost every ten minutes.

Q: What are the most useful PD programs for an Amiga graphics artist?

A: There are some image processing programs, but by far the most interesting are fractal generators. FracGen is a fractal pattern generator which builds complex pictures from simple two or three line shapes. A lot of man made and natural phenomena are do-able, and some examples of these are supplied, but the best fun is to be had experimenting with your own shapes, although getting ones which work perfectly is sometimes a little tricky. MandelVroom is a classic Mandelbrot creator, and one of the fastest around. The shapes can be zoomed in to reveal more and more detail.

One program which isn't a fractal generator is Rend24, a shareware program which loads a series of 24-bit, JPEG or GIF pictures and combines them into a HAM animation – a brilliant way to preview a 3D animation created with a 3D program or a fractal landscape editor. Once you have the sequence down as a HAM animation you can use the file as part of a piece of Amiga video, or how about using them as an intro to a game or program?

Q: Where can I get clip art for the Amiga in IFF, 24 bit IFF and JPEG formats?

A: The best place to find graphics is in the public domain, either through your local PD supplier or via a bulletin board system. On a BBS you can download graphics into your computer that have been created using graphics programs and digitisers and use them or treat them yourself as clip art in your DTP programs or for treatment in your image processing programs.

Q: What do the new AA graphics modes have over the old Amiga ECS

machines? Is it worth upgrading or is there a way I can do as good if not better with my old ECS machine?

A: Although at the time of going to press there is no upgrade kit available for the Amiga 500, 600 or 2000 to turn them into AA chipset machines, it's rumoured that one will be available shortly. The AA machines like the 1200 and 4000 have more colours available and you can also do HAM mode with interlace to double the vertical resolution (with a little interlace flicker of course).

Obviously if you add a 24-bit option to your machine, like an OpalVision, Harlequin or DCTV board, then you have better quality than the AA chips can offer, although you won't have a lot of the options available to Workbench 3 like the ability to load pictures into Workbench windows. If the rumours are correct Workbench 3 will be available to other users under the title Workbench 2.1.

JARGON BUSTER

AA – Sometimes called AGA or Advanced Graphics Architecture. A new graphics standard on the Amiga 1200 and 4000 allowing 256 colours on-screen in standard modes and 262,000 in extended modes.

anim – A compressed animation format, that stores the differences between frames rather than the whole pictures.

anti-aliasing – Smoothing method used in computers to blur the transition between sharp edges and background colours by using intermediate colours.

chip RAM – The amount of memory available to the custom chips, usually the first 1Mb (or in some cases 2Mb) of memory.

fast RAM – Expansion memory, up to 8Mb on standard Amigas, and more if the machine has an accelerator on board.

GIF – A compressed file format used by CompuServe.

IFF – Short form of Interchange File Format. All types of Amiga files are interchangeable between programs, like IFF ILBM files are pictures, IFF ANIM files are animations. Created by Electronic Arts and Commodore Amiga.

ILBM – Short for InterLeaved BitMap.

interlace – Doubles the vertical resolution by alternating two sets of 256 lines

very fast, giving you a slight flickering effect.

JPEG – A compressed file format used for 24-bit pictures, which are about a 16th of the original size in most cases.

Pixel – Coloured dots on the computer screen.

Rendering – The creation of graphics by calculation rather than by drawing or painting.

MONITORS AND TVs

Having an Amiga is all very nice, but how do you get to see all those lovely colours unless you have a good monitor? What is a good monitor? Can you use any old thing, or will only a very small selection of things do?

Choosing a monitor is partly a matter of personal taste, partly dictated by your system. In the beginning it's a straight toss up between the two simple RGB monitors generally teamed with an Amiga bundle by Philips or Commodore which can also handle composite video. The next steps up depend to a large extent on your Amiga, and whether it can handle any more display modes and send out anything other than the standard.

The spec of your Amiga, and what can be added to it, will open up a range of different options. By and large the older your system, the less choice you'll have. Even so you can still add the functionality of the newer Amigas to your machine, whatever its age. Let's flash through a barrage of the most frequently asked questions about displaying your Amiga through monitors and TVs:

Q: What is the best type of monitor for my Amiga? Is there one unit that will do all jobs?

A: The ideal first monitor is any analogue RGB monitor, like the Philips 8833 or the Commodore 1084. It's a good idea if your monitor choice has a Scart plug in the back, and stereo speakers for the Amiga's digital stereo sound. For flicker-free output to screen, you need something which can send a double scan rate and a monitor which can receive it. Add-ons which do this are available for all Amigas which don't already offer this facility.

The best kind of "one size fits all" monitor must be a SVGA multisync, although it depends on the type as to whether it'll work with your system. The Commodore 1960 is a good example of one which will work, as it was made

with Amigas in mind. They're very expensive though.

Q: Can I use the VGA monitor from my PC with my Amiga?
A: If you have a Amiga 1200, 3000 or 4000 then the answer is yes, right away. These machines have the extra hardware and software to enable a 31kHz scan rate, meaning that VGA displays (usually destined for the PC clone market) can be employed to deliver flicker-free high resolution modes.

The display enhancer hardware is available as an upgrade in the form of the Commodore A2320 display enhancer board. This fits into your video slot (if you have a 1500 and 2000) creating the necessary scan rates for the higher speed monitors. The AGA chipset machines do all this kind of thing in software.

Q: What is the idea behind a so-called "multisync" monitor and can I use one with my Amiga?
A: A multisync monitor is one which automatically adjusts its scan rates to the incoming signal, so for example you can use one on a PC to cope with a number of different resolutions and scan rates, switching automatically as each one arises. The Amiga's lowest scan rate is 15kHz, and so to be really useful the multisync display of choice will have to be able to go down this low. If you have a display enhancing circuit (or software) in your Amiga then this is not a problem because the display will be pumped out at a much higher rate.

Q: I have a portable TV set. Can I use it as a monitor with my Amiga?
A: The majority of TV sets accept signals from the outside world via the aerial as RF or radio frequency signals. Therefore in order to show up on your TV screen, your Amiga must have an RF modulator to send a signal that the TV's aerial socket can interpret. Although some of the new Amigas have RF modulators built in, the older units like the 1000, 500 and 2000 need the external Commodore A520 RF Modulator in order to send the signal to the set.

Another method is if the set has a Scart RGB input socket in the back like a regular RGB monitor. If this is the case you can feed the signal from the Amiga's RGB socket directly from the RGB out into the back of the set, bypassing the RF circuit and giving a you a result not unlike that from a conventional monitor (you don't have to even change channels, as the Scart socket will often interrupt any channel on the TV side as soon as the Amiga is switched on and a signal is detected at the socket).

Q: I'm worried about harmful effects from my monitor screen. What precautions can I take?

A: Apart from putting a lead bag over your head, very little. Seriously though, the main hazards to health from working at a monitor for long periods are eyestrain, back problems, high levels of static electricity and low levels of radiation. These effects can be minimised by taking regular breaks from the screen, and ensuring that the lighting in the room is not glaring off the screen, the sharpness and brightness of the screen are unimpaired, and the chair you use is comfortable.

Screen filters help banish fuzzy screens and increase the contrast making the definition of the screen much better. Anti-static filters are also available which have a grounding strap which can be fixed to a nearby radiator or other grounding point. Filters also usually have an anti-glare coating, like spectacles or expensive binoculars, holding down any extra eyestrain.

Using an ioniser in your computer room can also help keep down the static and dust levels in the room, and will decrease the damaging effects of the high static charge on the air quality if not your body. Comfortable chairs with back rests are a very good idea, and some people recommend those funny Z-shaped chairs you sit on backwards (although they just seem to hurt your knees rather than your back as far as we can see). Good posture while at the keyboard is a good idea, and a book on Alexander Technique probably wouldn't go amiss. There are, incidentally, new EEC guidelines covering working in front of video displays.

Q: I have an old Commodore 64 monitor, the 1081. Can I use this with my Amiga, or is it incompatible? I'd like to run a video through the Scart socket and the Amiga with the TTL socket on the back of the monitor.

A: There's no need to use the TTL input (especially as this would limit you to 16 colours) as there is a Scart input on the monitor too. The ideal solution would seem to be to employ a Scart plug switcher (costing about £12) and plug both the video and the Amiga into the same hole. This gives you the best quality from each input and doesn't confuse the monitor by having too many sources coming in from different directions.

Q: I want to buy a 24-bit graphics card but I'm not sure which to buy. Do I need an external monitor to run the card or can I get something which works on my standard Philips 8833?

A: Some graphics cards like the Harlequin need their own monitor (additional to the one you already have on your Amiga) to display 24-bit graphics, and this can be anything from a standard RGB monitor (like the Philips 8833) up to a multisync. A lead goes from the back of the card to a Scart plug at the back of the extra monitor.

Obviously you could get a Scart switcher box which lets you put two Scart

feeds into the one monitor. You have a switch to flip between the two inputs, saving you the trouble and expense of buying an additional monitor. The DCTV (Digital Creations) needs an external composite monitor, which could just as easily be a portable TV or video with a Video In socket.

OpalVision (Opal Tech) however creates a second channel on your screen for 24-bit graphics, and enables you to use your standard monitor with no need for manual switching when it uses 24-bit mode. The monitor acts normally in all other modes and no adjustment is necessary. You can even load a picture into the 24-bit buffer on the card to sit behind your Workbench!

Q: I have an old monitor which has three or four bayonet fit plugs on the back (designed for what I don't know). When I connect the Red Green and Blue signals from my Amiga to it using an adaptor lead, the image is there but it's all scrambled. Do I need extra hardware to use this monitor?

A: The bayonet plugs are BNC connectors, much used in workstations and high-end video work. The probability is that the monitor is usable, but it's just too fast a scan rate for the Amiga to match. It may be VGA or SVGA which have scan rates of around 31kHz rather than the standard Amiga output of 15kHz. ECS machines can cope with higher scan rates, for example the Amiga 3000 has 31kHz output, as does the AA chipset Amiga 1200 and 4000, so trying the monitor on one of these machines if you know anyone who has one is definitely a first step, as this will confirm at least one clue to the monitor's display standard.

If it is a higher resolution monitor like VGA, then you can add the Amiga A2320 display enhancer card to the video slot of your 2000 or the ICD flicker-free unit to your 500/600. This ups the output of the display from the Amiga to match that of a VGA style monitor, and enables you to use the higher scan rate monitor. But before you make any outlay on a new display adaptor, confirm or deny the true status of the monitor by trying it on a computer which can show this rate. If it's a rate other than the VGA rate, then you may need to get a monitor driver for your monitor drawer to drive it. It may not be possible to drive it properly, but all you can do is test out the possibilities.

Q: I've bought a second-hand monitor at an auction. How can I find out what the pin-out connections on the plug are so I fit it to my Amiga? It's a very odd plug type I've never seen before.

A: We have to ask why you bought this monster in the first place? The answer to your question is that unless you're an electronics whizz you can't find out what the pin-outs are without grave hazards to yourself and your machine.

Buying stuff at auctions is an exact science, and unless you know exactly what it is you're buying it's a very hazardous business. There have been a

number of failed display formats and types of monitors designed for dedicated systems at very exotic scan rates, which are useless unless you own the machine for which the display was made (and the date happens to be 1974). The best thing to do is only go for stuff you recognise, and even read up on old-fashioned equipment to get to know what is compatible and what isn't.

RGB monitors are good, Scart plugs are fairly new and so a dead giveaway that you might have a chance of running the thing. Know your onions, that's the best advice. And forget trying to attach that old monitor to your system. Find out about it from the firm which made it, get them to send data on it (if they're still trading) and eventually you might discover it's VGA or SVGA or something. You can use such a monitor but you need an extra piece of kit to attach it to your Amiga (see question 2 and 8).

Q: What are the Euro72, Super72, DblNTSC and DblPAL modes spoken about in the new AGA machines?

A: The Euro72 and Super72 display modes are variations on the VGA usual modes. The other two are monitor drivers in the Monitors drawer which are to do with the "scan doubling" feature of the new machines which allows each scan line on the display to be shown twice. These new modes do in software what the display enhancer does in hardware in the A3000 and the A2320 card. With the so-called "promotion" feature enabled, the displays are "promoted" to a flicker-free version. These modes are also variations on the VGA monitor but are designed to fit the PAL and NTSC dimensions, but at twice the refresh rate (to make them usable on a VGA or multisync monitor). So DblNTSC will provide a 640 x 200 non-interlaced screen, and DblPAL will do 640 x 512 non-interlaced. Non-interlaced screens will take up the same amount of space through scan doubling.

Q: What is the best method of display, RGB, Composite, RF modulator, or Scart?

A: Scart is a sort of plug, which can carry RGB or even composite signals. RGB is the clearest method of displaying a screen from your Amiga, as this is direct and pure colour from the computer to the screen. Composite video is the next one down the scale of clarity and quality, as this is a direct video signal combining all three of the red, green and blue components on one cable, at the loss of some quality.

Finally we have RF modulation, which because it is converting the RGB signal to a radio frequency loses even more quality. The best form of transmission of a computer picture to a display is RGB, which is why smart TV sets and videos have Scart plugs to take the picture directly to the display and cut out all the fluffiness of RF modulation.

Q: I've got access to an NTSC version of the Commodore Amiga monitor. Can I use an NTSC monitor with my Amiga?

A: It's not so much that you can't display anything on an NTSC monitor, you can in fact, albeit lacking the last 56 lines of a PAL picture. The problem is that the disparity between the mains frequency of the UK and USA. In order to operate the monitor you will need some form of transformer to downgrade the frequency of the power flowing into the unit. It's less than useless, unless you intend to use a device which is NTSC only, like the NewTek Video Toaster, and all the other NTSC gear that that entails. Then an NTSC monitor wouldn't be out of place, because you'd have to use all your equipment at the US mains frequency.

Q: Is there anything I can get which will allow me to use my monitor as a TV?

A: Yes, a TV tuner, a number of which are available on the market, will supply the ability to tune into the TV channels and feed them to your monitor. You can also use your video machine as a tuner, as it has one built in. For example if you have a Philips 8833, take a lead from the Video Out port on your video to the "colour composite in" on your monitor. Then all you have to do to switch between computer and video is click the Input I/II switch on the front of the monitor. Or you can just take a Scart lead out from the video and stick it into the back of the monitor.

There is a third option, and that is the RocTec PIP View. This is a sophisticated TV tuner for use with monitors which enables you to view your TV picture as a small window on your Amiga screen (you can also show the Amiga picture inset into a TV picture).

JARGON BUSTER

multisync – A monitor capable of adjusting automatically to a variety of different scan rates.

NTSC – The US and Japanese TV standard, consisting of 525 lines running at 30 frames per second. The letters stand for National Television Standards Committee, although it's always jokingly referred to and Never Twice the Same Colour, due its slightly fluffy handling of colour (see PAL).

PAL –The UK TV standard consisting of 625 lines running at 25 frames per second. The letters stand for Phase Alternation Line

PIP – Short for Picture In Picture. A smaller version of one TV channel inset

into another.

RF – Short for Radio Frequency. If a signal from a source is to be fed into a TV aerial socket, it needs to be fed into a TV modulator first which creates a signal at the correct radio frequency to be read by the TVs tuner. There is a considerable loss in quality transmitting a pure RGB signal to a screen using this method.

RGB – A pure video signal composed of the Red, Green and Blue components of a colour video picture. The Amiga outputs RGB from the socket usually used to connect the Amiga to a monitor.

scan rate – The screen refresh rate of a monitor, a normal RGB monitor has a 15kHz output, a VGA monitor 31kHz.

Scart – A large, square 21-pin connector often found on the back of TV sets monitors. Is capable of sending RGB, audio, and video signals down the same cable.

tuner – A video tuner, like a radio tuner, captures the radio frequency waves from an aerial and allows you to tune into channels on the carrier waves. In the case of a TV tuner all the UK TV channels can be tuned into and seen on a monitor if you have either a stand-alone TV tuner or use the tuner in a video recorder.

VGA – A 256 colour 640 x 400 graphics standard from the PC world, which is rapidly finding a home on the Amiga, meaning that VGA monitors can be used on Amiga 1200, 3000 or 4000. VGA displays can be used on other Amigas with the addition of display enhancer hardware.

MUSIC

Getting into Midi and sound sampling on the Amiga nowadays just couldn't be easier. There are music programs and sampling hardware to suit all pockets and even Midi equipment, such as synthesizers and drum machines, which used to be quite expensive, have come down in price in recent years.

With this new technology, though, has come the inevitable learning curve problems and as the number of sequencer and sampled sound users have grown so, needless to say, have the number of questions on Midi and other related music topics...

Q: Everyone says that sequencers and Midi have made music easy but surely

you still need to be able to master a piano-style synthesizer keyboard in order to do anything useful with a sequencer?

A: You need to be able to find your way around a synth keyboard but you definitely do not, by any stretch of the imagination, need to become proficient. In fact by using only one or two fingers, and sticking to the "white note" keys (C and A minor) that are easy to play in, it is possible for most people to pick up the basics in a few hours (you can always transpose songs to their required keys when they are finished).

This may be a cop out in a sense but even if it is you are in good company – many professional musicians who are not primarily keyboard players adopt exactly the same approach simply because of their need for Midi facilities.

Q: Having written a lot of sequenced music over the last few years I'm now beginning to find it awkward to remember instrument/channel settings for all my compositions. Why can't sequencer programs let users add text notes to sequence files to hold these types of details?

A: Some do and Dr T's KCS is one example. The best idea however is to make life easier by standardising the use of your Midi channels so that you get used to seeing particular instruments on particular channels (eg use channel ten for drums, 1 for bass, 2-6 for any synth modules, 11-16 for digital delay and other effects units etc).

These sort of conventions will make it easy to recognise sequence data, prevent silly mistakes being made, and eliminate the need to keep altering the receive/transmit channels of your equipment.

Q: I love music and can even knock out a tune on a synthesizer keyboard. When it comes to programming Midi drum parts however I am totally lost. Is it worth persevering?

A: Persevere by all means but don't get hung up about it. Instead do what most other people do faced with the same situation – cheat. There is plenty of non-copyright material around in the public domain and lots of synths, sequencers, and drum machines come with their own pre-programmed drum patterns. There is of course nothing to prevent you from borrowing ideas and drum pattern phrases from the examples you find. Alternatively there are a number of music companies who nowadays will supply libraries of sequenced drum patterns in Midi file or other sequencer-readable formats.

Q: I'm stuck with a synthesizer which seems to have more than its fair selection of weak sounding preset voices. Is there a way of thickening up the sounds using my sequencer?

A: Doubling, where you take a track, duplicate it, and then change the Midi channel of the second track so that two different voices play essentially the same notes, can usually help. Shifting the start time of one of the pairs of tracks should further improve things.

Q: I recently changed my synthesizer only to find that the program change events in my existing songs no longer correspond to suitable voices on my new synth. Is this likely to happen next time I change my equipment as well?
A: Possibly, although the new Roland GS standard is designed to eliminate these sorts of problems in future generations of Midi equipment. One solution when buying expander modules and the like is to opt for units which have configurable program change tables so that you can set up the program change numbervoice correspondences to suit your existing sequences.

Another trick is to use short introductory sequences to set up your Midi instruments rather than embedding all program changes in the early parts of your main song. A lot of Midi musicians use introductory one-bar sequences containing a count-in (eg a high-hat playing once on each beat) plus the program change events required to set up synthesizers, drum-machines, delay units and so on. Such control sequences are easy to edit because they are short!

Q: My Midi songs seem to take up an awful lot of memory and disk space and this in turn prevents me from holding more than two or three songs in memory at the same time. Would buying more memory help?
A: Yes, but the problem might be that you are recording long songs as a single sequence when, due to repetition of particular parts, this might be unnecessary. With a lot of pop songs for instance it is usually possible to build the various parts of the song by recording a beginning, an end, plus the appropriate verses and choruses. Having done that the final song can be built by linking the various sections together.

This approach will mean that even if your song plays ten verses and ten choruses you'll only actually store the Midi data associated with a single verse and chorus. As well as saving space this would also make it easy to alter the verse/chorus format arrangements of your songs.

Q: Is the arrival of 16-bit sound sampling packages, such as Clarity 16, going to mean the death of 8-bit Amiga sound sampling?
A: Despite what some pundits say, the demise of 8-bit sampling is fairly unlikely for two reasons. Firstly, 16-bit quality is simply not needed for many Amiga audio applications. Secondly, 16-bit samples take up an awful lot of memory and when users realise that budget packages such as Clarity 16 do not

providing direct to disk recording/playback many may change their mind about needing 16-bit quality anyway!

Q: I've started trying to write some simple Midi programs using Basic with Print statements being used to send data to the serial port. How can I prevent note control character sequences from making my code look untidy?

A: The best idea is to isolate the control sequences so that the main body of code is not littered with awkward-to-read statements. Suppose for instance that you want to generate Midi data by opening the serial device as a file and using these types of Print statements...

```
PRINT# 1,CHR$(&H90 OR
channel)+CHR$(note)+CHR$(velocity);
```

Rather than embedding these types of control sequences in your main code, which causes programs to look messy, why not isolate the sequences into separate subroutines or functions? For the above example a user-defined function could be defined like this.:

```
  DEF FNNoteOn$(note,ch)=CHR$(&H90 OR      -
ch)+CHR$(note)+CHR$(velocity)
```

which in turn would enable Midi note-on data to be sent using the more readable expression:

```
PRINT#1, FNNoteOn$(note,channel)
```

Q: What's the difference between pre-record and post-record Midi sequencer filtering and which is best?

A: Pre-record filters take out, or modify, the Midi messages before storing them as sequencer track events, so by the time a sequence has been recorded those changes are permanently stored as part of the track data. Some sequencers are also able to produce filter and conversion effects after the data has been stored. These are post-record options and here the effects and settings work on the output side of the sequencer (Bars & Pipes incidentally is brilliant as far as this particular area is concerned).

The big difference of course is that these "output only" filters do not prevent the original Midi messages being stored as track data. If, at a later date, the filter option was cancelled any messages previously being suppressed by the filter would again become part of the sequencer's output stream. Post-record options are usually best because they have the benefit of flexibility since you can always undo a particular setting. Pre-record filter options however are

useful in that they can eliminate the storage of unwanted Midi information.

Q: Is it true that quantisation can destroy the "feel" of a piece of music?
A: Quantising all tracks to complete timing perfection can certainly take the sparkle out of a performance but over the last few years a lot of effort has gone into finding suitable partial quantisation schemes in order to prevent the feel of a piece of music being destroyed.

One option is to tidy up the notes a bit without making the timings 100 per cent perfect. Another is to only quantise those notes which lie very near the hypothetical quantise division lines. Some sequencers offer random timing improvements to prevent all notes being quantised to the same degree.

Implementations vary. Gajit's Sequencer One for example allows the beginnings of notes to be quantised whilst keeping the note endings unchanged. Tiger Cub allows the first note in a bar to be fixed (quantise protected) and then allows the quantisation of subsequent notes to be made relative to the first note. It also allows variable offsets to be added so that a "swing" feel can be introduced. As you go higher up the scale in sequencer power you find more and more sophisticated quantising schemes being made available.

Q: What are RAW sound samples, how are they used, and how do they differ from IFF sound files?
A: When you record a digital copy of sound the sound sampler measures the sound waveforms being presented as input and stores them in memory as a series of numbers. This digital form, which is the digital equivalent of the original sound, is called the raw waveform data. RAW sound sample files are just binary files containing this digitised waveform data (many games programmers opt for using raw sound samples and include the details of playback rates and so on, within their programs).

IFF files contain the same sort of waveform data but in addition to this they contain type and playback details that can tell the program reading the file how the sound should be played.

Q: How long can a Midi lead be? I'd like to try remotely controlling my band's Midi gear from off-stage!
A: The Midi standard itself says 15 metres but a lot depends on the amount, and the peak densities, of the Midi traffic flowing through it. In some cases leads of twice the maximum suggested length have been found to work. The number of pieces of equipment being used may become an important factor because each time the Midi signals pass through a unit there will be a further slight deterioration in signal quality.

If, due to long cable pathways, the signals are poor to start with it is possible that you might experience difficulties with units at the end of your equipment chain. If you are thinking of experimenting it might be worth terminating the long lead with a Midi Thru box so that you can star network the Midi units to minimise any signal degradation problems.

If your initial efforts are unsuccessful you might like to know that Philip Rees (0608 811215) now produce special line driver units that effectively remove the 15-metre Midi cable limit altogether. They'll cost about £90 a pair but your band might prefer to lash out on these rather than have the worry of whether an over-length lead is going to play up in the middle of a gig!

Q: For general Midi work synthesizers need to be polyphonic, ie able to play many notes at the same time. But why is it useful for the synthesizer to be multi-timbral, ie capable of playing more than one sound at the same time, as well?

A: It's because multi-timbral synths can be programmed so that different voices respond to different Midi channels. You may set up the synthesizer so that data received on channel 1 is played with a violin voice, data received on channel 2 is played using a bass-guitar sound, and channel 3 data gets played using a oboe sound. Having recorded a number of tracks on the appropriate Midi channels you would, when playing back all the tracks at the same time, be able to hear all the different instruments playing simultaneously!

Q: What is the difference between Midi status bytes and Midi data bytes and how does a piece of Midi equipment tell the difference?

A: Midi sends its information in eight-bit units. As you'll no doubt know the computer world calls these bytes and a byte can represent a letter, a number, a computer instruction or anything else – providing it can be coded as one of the 256 patterns which an eight-bit binary number can represent. Midi messages then are streams of eight-bit numbers and it is the Midi standard which has defined their meaning.

Some Midi messages can consist of more than one byte. The first byte, known as the status byte, identifies the general message class. Any other bytes of the message that exist are called data bytes.

How does Midi distinguish between status bytes and data bytes? It uses the uppermost bit of each byte – status bytes always have the high bit (bit 7) set so these numbers can range from decimal 128 to decimal 255 (10000000 binary to 11111111 binary). Data bytes are therefore restricted to values ranging from decimal 0 to decimal 127 (00000000 binary to 01111111 binary).

Q: Since Midi is supposed to be a communications standard why does it

allow the use of messages which are non-standard, ie SysEx messages?
A: Because it gives manufacturers of Midi equipment the chance to implement their own specialist control functions. Nowadays there is a trend towards implementing all manner of voice modification and remote front-panel control using SysEx messages. It is in fact these types of information packets which allow things like patch editors and librarians to perform their magic.

JARGON BUSTER

drum part – This used to mean the part of a music score containing rhythm and drum/percussion note details. Nowadays Midi musicians take the same term to mean the track/tracks on which drum and percussion events are stored.

drum machine – An electronic unit which can generate rhythms using either or sampled drum sounds. Nowadays most drum machines can generate, and be driven by, Midi note data.

patch editor – A program which allows the user to create, edit, store and load synthesizer voices using their computer rather than the front-panel facilities of the synthesizer itself.

program change – A Midi message used to tell a synthesizer or similar unit to change to another voice or program setting.

transpose – To change the key of a piece of music.

quantisation – The automatic adjustment of the timing of a series of notes in order to make them play exactly on particular divisions of the bar.

PRINTERS

Most printers are based on what is essentially quite simple technology. Why then should they be the subject of such confusion? The simple answer is standardisation; or rather, the lack of it. The vast majority of printers connect through one of two ports – serial (RS232) and parallel (Centronics). Some computer systems make use of another port known as SCSI, but since this is not standard fitting on the Amiga, no manufacturers have made use of it. SCSI is extremely fast when compared to either parallel or serial ports (hence it's almost the exclusive preserve of hard disks) but even the fastest printers are nothing like quick enough to keep up.

Very broadly speaking, printers divide into two categories according to how

the characters are formed: dot matrix (graphic) and character; and of these, dot matrix make up the vast majority. Character printers are the grand-daddies of them all – and have changed very little in the years since they evolved from simple typewriters. All character printers have one thing in common; the characters are pre-formed on hammers which strike an ink-drenched or carbon backed ribbon against a roller.

By sandwiching a piece of paper between the roller (more correctly called the platen) the characters are formed. A sub-division of this system is the golf ball. In this case the characters are all located on a single ball which rotates and tilts to face the required character against the paper. The advantage of this odd-sounding system is the golf ball "hammer" can be easily changed to select alternate character sets. Such mechanical systems are very limited in terms of speed and very, very loud in use!

This system evolved the next generation of printers which still use hammer-against-platen printing: impact dot matrix. Dot matrix has several advantages over the more classical approach – not least that it is possible to produce graphics at very high resolution. In an impact dot matrix printer the head contains 8,9 or 24 electrically operated hammers – or pins. As the carriage (print-head carrier) travels from left to right (and right to left for bi-directional printing) the characters are constructed one vertical row at a time by "firing" the pins.

The exact timing is critical, but is relatively simple for modern electronics, and adjusting how fast the characters are formed allows the dot matrix to adjust the character pitch. The number of pins governs the printer's physical resolution and character quality – the more the better.

Time and technology wait for no man though, and developments have lead to great improvements in print-head technology. The hammer's solenoid coils have been replaced by heaters which can heat a tiny amount of ink and spit it out of minute nozzles like an aerosol. This "bubblejet" technology is the quietest form of printing yet devised. It also delivers very high quality output since the ink is deposited directly on the page, although special paper and inks are required which do increase the cost of the printed page.

However, the story of dot matrix does not end there. Optical printers have been devised which use light-sensitive rollers similar to those found in laser printers and photocopiers. Although commonly called laser printers, these printers are more correctly termed LED page printers: the acclaimed Oki 400 is a good example. In these printers, characters are formed by an LED array which replaces the conventional print head and carriage (the array is the full page width) and operate by dumping a raster (line-by-line) image on the page.

The advantage of these printers is the consumable items are much cheaper than comparable lasers or even bubblejets! Also the quality is at least as good as "real" laser printers: and less moving parts means they are inherently very reliable. The heat-transfer system (just like a photocopier) also means these

printers get very hot inside and a large fan is required to keep the electronics cool. Therefore these printers are often more noisy than a simple bubblejet.

Q: I have an old Epson printer which seems to squash lower case letters. Is there any way to fix this?

A: Not directly. This type of printer is an early 8-pin design and is incapable of producing true descenders – that's what the extra pin is used for. One solution would be to use a word publishing program such as Wordworth 2 or Final Copy 2 – this will drive your printer in graphics mode and override the internal fonts. However, it is doubtful the quality will be very good.

Q: When I print in colour from Deluxe Paint, the output gets smudgy – especially on the lighter colours such as yellow. Also the blue takes on a greenish tinge – what am I doing wrong?

A: This problem affects all 9-pin impact dot matrix designs such as the Star LC-10C, LC-100/200 and Citizen range.

The problem is that you have selected a density which has placed the printer in double or triple pass printing. Workbench does not give any warning of this but it is detailed in the manual.

When you select high density printing from the preferences, the printer gets more dots per inch by printing in almost the same position two or three times. This means that a dark colour, say red or blue, can get back onto the yellow part of ribbon.

The only solution is to replace the ribbon, change to the lowest density and start from scratch, learning from this all-too-easy mistake.

Q: I have seen refill kits advertised for the cartridge on my HP-Deskjet. They seem very cheap – but are they worth it?

A: Generally speaking, no, and there's a very good reason for this. Inkjet printers use a special type of ink with fast evaporating solvent and you can't guarantee the replacement you get will be of the right quality. Moreover, when you replace the cartridge, you are also replacing the entire head and therefore always ensuring your machine performs at its best. Inkjet consumables are costly – the price you pay for good quality output.

Q: I have a Canon bubblejet printer and cannot seem to get the near-laser quality people talk about – the internal characters seem very blurred. Is there anything I can do about this or have I just been conned?

A: This problem crops up time and time again and is most often not a fault

with the machine, but the paper! Paper is made from wood pulp and as such tends to have a grain – look closely at a sheet of newspaper and you'll see this. The ink used in any inkjet is very thin since it has to travel down very tiny tubes and form little bubbles, say 1/300in across by the time it hits the paper! Rough grain paper such as that usually sold for computers is not suitable because the ink tends to run down every little nook and cranny – just as rain spreads over a paving stone. This is the blurring effect you have experienced – and can be fixed by using better quality paper.

A coated laser bond like Courier Laser Master should do the trick – but be careful you use the right side! It looks shiny and feels very smooth.

Q: I have Workbench 3 and cannot get my printer driver to appear in the Printer Preferences window as described in the manual. Is there any way to fix this?

A: This is a bug in Workbench 3 and only crops up when you use it on a single-disk drive machine, such as an A1200! The fix is quite simple to achieve although it does require some work with ED. Before doing anything else, make a copy of your Workbench disk and boot with that.

1. Select Execute Command from the Workbench menu.

2. Enter the following in the window:

```
ED S:Startup-sequence
```

and click OK.

3. If everything goes to plan, ED will start up and load the Startup-sequence (that's a file on the Workbench that gets executed every time the machine is booted). If ED comes up with a blank screen, select QUIT from the Project menu and try again from Step 1.

4. Now press the down-arrow on your keyboard until the blue block (cursor) is at the start of the line:

```
ASSIGN >NIL: PRINTERS: DEVS:Printers
```

5. Hold either Shift key and press the right arrow key. The cursor will move to the end of the line.

6. Press the spacebar once and enter the word PATH. The line should now read like this:

```
ASSIGN >NIL: PRINTERS: DEVS:Printers PATH
```

While in there, you may also find it handy to perform the same change on the line:

```
ASSIGN >NIL: KEYMAPS: DEVS:Keymaps
```

7. Select Save from the project menu, wait a few seconds for your changes to get written to disk. Now re-boot the machine and check everything is working all right – including the Printer Preferences. If something has gone wrong, re-copy the Workbench disk and start again.

Q: When I produce graphics from Deluxe Paint, I get a black stripe along the top and right edge of the screen. Is this a fault on the printer?
A: This is a surprisingly common complaint with Deluxe Paint – it has to do with the screen page size being slightly larger than the display. The simple remedy is to select a background colour and use the CLS gadget from the toolbox. A useful precaution is to clear the background bright white – and work on that. This will ensure you do not wear out the ribbon unnecessarily.

Q: What causes the horizontal stripes across the page when I print from Deluxe Paint, Wordworth and PageSetter II?
A: Sadly this is a feature of dot matrix printing and has to do with the way the printer shifts paper up past the print head during graphic printing. It is possible to improve the output slightly by using single sheet paper instead of tractor feed since this is inherently more accurate. Some manufacturers, Citizen for example, produce their own high-quality drivers which will improve matters over the supplied general purpose ones supplied by Commodore.

Q: My friend has a HP Laserjet. Is there any way I can send the printed output from my applications to a file and print them on his machine?
A: Some wordprocessors and databases can send output to a file from the print requester, although this is not supported on low-cost packages. You'll find a natty little utility in the Tools drawer called CMD which can do this for you automatically. Although CMD is best used from CLI, it can be operated from Workbench. All you have to do is double-click it and let it do the rest.

Everything the application sends to the printer is automatically sent to a file. If you have Workbench 2 or higher, you can copy the file directly from the RAM disk otherwise you will have to resort to CLI or a disk manager program like SID or Directory Opus. The file can be sent to a printer using AmigaDOS

like this:

```
COPY MyFile to PAR:
```

Note: you must use the PAR: device since this is sent directly to the printer rather than pre-processed by the printer driver through PRT:. You can use this technique to print files on other machines such as the PC by sending the binary file to the PC's printer via MS-DOS.

Q: I don't have a printer attached to my machine. However I sometimes find myself activating the Print requester in some of my programs. This locks them out and eventually pops up an annoying message telling me the printer I don't have is not responding! Any suggestions?

A: One I have experienced many times when the printer is switched off or packed away from prying pinkies... One possible solution is to add the following line to your Startup-sequence before LoadWB:

```
RUN  NIL: CMD -M Parallel NIL:
```

The -M options forces CMD to run until you stop it – generally speaking, you don't need to until you get a printer. Alternatively, you can set these options from the Tooltypes array viz.:

```
FILE=NIL:
MULTIPLE=TRUE
```

Note the ToolTypes array is case sensitive – so Multiple=True would not work. Also, don't forget to double click the icon before you start any applications.

Q: What is the difference between a ribbon for my 24-pin printer and the same make and model in a 9-pin? The ribbons for the 9-pin machines seem a lot cheaper...

A: The problem is the pins on your 24-pin machine are much smaller, and therefore, much more brittle than those on the 9-pin models. To complement this, the manufacturers make the 24-pin ribbons from a thinner material which is less likely to damage the head. Given the price of a replacement head, cheap ribbons are often false economy.

Q: If the Amiga is a multitasking machine why can't I print from two applications at the same time?

A: Sending output to the same printer at the same time would be senseless –

assuming it were possible. In theory it is possible to print on a parallel printer and serial printer (perhaps a PostScript laser) at the same time, although I have never had opportunity to try this.

Due to the way the Amiga shares its devices, it should be possible to print from two different applications: while one is processing the other is printing and vice-versa. Just occasionally though, applications grab the printer for their own use: and worse still refuse to give it up when they quit! If this happens your other applications will be unable to access the printer and the only solution is a complete system re-boot.

Q: What is an Esc code?

A: The Esc character is a standard Ascii character (27) called "Escape". Many printers use this code to mark the start of an internal printer command. Esc is followed by one or more numbers making up the command and its parameters, rather like programming in machine code. For most purposes, the escape codes are generated by the printer drivers supplied with your machine. If you do not have a suitable driver, you'll have to make do with Generic – which is a simple text-only filter.

Q: I have Workbench 1.3 – is it possible to have more than one set of printer preferences available so I can choose the ones I require for any particular job?

A: This is possible in theory although it requires a certain amount of fiddling with AmigaDOS – and you don't want to get into that unless it can be avoided. The best, although not perfect, solution for this problem is to keep several copies of Workbench; never a bad idea in itself.

You can have one configured for high-quality mono graphics printing; another for draft colour graphics and another for text use. All you have to do is boot with the one you need.

Q: I have an Oki page printer and the characters seem smudgy around the serifs. What can I do?

A: Clean it. This sort of problem seems to occur when the LED array gets dirty. You should also clean the fuser wire – but wait until the internals have cooled down, use the supplied tools and above all do it carefully!

JARGON BUSTER

baseline – The invisible line all characters appear to sit on. Remember when you learned to write in pre-school? The rules you used are the direct equivalent of the baseline.

Centronics – A little-known printer manufacturer who devised the most widely used parallel printer standard in use today.

CPI – Characters Per Inch. The number of printed characters per inch of available line space; also called character pitch. The most popular three name pitch settings are: Pica (10 CPI); Elite (12 CPI) and condensed (17-20 CPI).

CPS – Characters per Second. The number of letter characters printed in one second. Caution: this figure is usually optimised by using a small character which prints faster.

consumable – Anything a computer uses which much be replaced. Examples include paper, ribbons, ink-cartridges, toner cartridges and so on. These determine the running costs of any device. Don't forget electricity though...

descender – The part below the baseline of lower-case letters such as "g", "p" and "y".

HPGL – Hewlett Packard Graphics Language. A printer graphics standard pioneered by Hewlett Packard. HPGL is easier to implement than PostScript and is emulated by a large number of low-cost bubblejet and laser printers.

MTBF – Mean Time Before Failure. The average usage time in hours taken before complete failure of a major internal component.

MTTR – Mean Time To Repair. The average time in hours to replace a part and get the system back on line.

NLQ – Near Letter Quality. Output from the printer is adjusted so the dot matrix is almost invisible and it looks like the work had been typed or produced with a conventional impact system.

PostScript – A page description language developed by Adobe – a very small and very rich American company. What's so special with PostScript is that the work is done by the printer – the computer only has to send a resolution-independent PostScript program. This means a operator can check his typesetting on a low-cost PostScript laser and get the final output produced on "bromide" at ten times the resolution: without changing a single parameter! Several attempts have been made to knock PostScript from its perch but none has yet succeeded.

PPM – Pages Per Minute. The throughput of a laser or LED page printer.

RS232 – The almost laughable "standard" devised for serial communications between computers or a computer and other small devices. Made more famous by ITV's Spitting Image's satirical TV advert.

SCSI – Pronounced "Skuzzy", this is an abbreviation of Small Computer Systems Interface. SCSI is effectively a modern, high-speed parallel port which has a language all of its own. Up to seven devices can be connected in parallel to the SCSI bus the computer can address each one individually. Typically only used for non-PostScript printers on the Macintosh system; although the Amiga could use them in theory if someone wrote a driver.

serif – Little flashes on letters. Times is the classic serif font; whereas Helvetica is sans-serif (without serifs).

PROGRAMMING

Amiga programmers have quite a choice when it comes to languages – APL, ARexx, Basic, C, C++, Forth, Fortran, Modula 2, Pascal and even Lisp and Prolog are among the possible choices. For serious Amiga programmers C is the most important language simply because almost all of the Amiga's technical manuals assume you understand it.

Most of the questions which follow concern either Basic or C and this is simply a reflection of the interest that exists in these two languages. Interest is now also growing in ARexx and, somewhat surprisingly, even the quite difficult job of programming the Amiga at the 68000 assembler level is attracting additional followers. What sort of things do readers commonly ask? Read on and find out...

Q: What computer language is best for beginners wanting to learn how to program the Amiga?

A: Almost everyone starts by using some form of Basic as their first language. As far as facilities go Amos is undoubtedly good but in the early days there is a lot to be said for sticking to a Microsoft-flavoured version such as HiSoft Basic. Why? Microsoft have set an established standard in the computer world and most of what you learn about a Microsoft-style Basic will stand you in good stead not just with Basic on the Amiga but with Basic on many other computers.

Q: Guru Numbers – What do they mean?

A: Guru Meditation Numbers are codes usually displayed (using an Alert

requester) just before the machine is about to crash. The codes are divided into several parts with the bit to the right of the decimal point representing the memory location of the task running when the error appeared.

The left-hand portion is an encoded error number in which the first two digits identify the operating system module that reported the error. The next two digits indicate a general error class (used to identify such things such as out-of-memory conditions) and the last four digits give additional error-class specific information.

Gurus can also be caused by problems detected by the 68000 processor itself. These are called processor exceptions and if this occurs the Guru subsystem and general error codes will be zero. The exception Guru which starts with 00000003 represents a specific error code of 03, and this is probably the most familiar to most Amiga users. It is caused by a 68000 program instruction trying to access an odd-numbered memory address when it shouldn't (this produces so called addressing errors).

It's worth mentioning that when the initial "Software Error – task held" requester appears it usually means exactly that – the task has just been prevented from running. If you've got files in RAM that you do not want to loose then don't select CANCEL straight away! Providing that the program concerned has not destroyed part of the operating system (eg by overwriting important system memory locations) you can usually to switch to another window (or open another CLI) and save any important files before selecting Cancel and rebooting!

Q: Most people know that FD files on the EXTRAS disk can be converted to bitmap (.BMAP) files to be used as libraries in Basic, but what else can they be used for?

A: FD stands for Function Description and the files provide details of the library functions and the 68000 processor registers which must be used to hold the parameters (arguments) passed to the function. If, for instance, you were to take a look inside the graphics_lib.fd file you would see definitions like this:

```
MakeVPort(view,viewPort)(A0/A1)
LoadView(view)(A1)
```

In this case MakeVPort(), and LoadView() are graphics library function names and the A0 and A1 references are the microprocessor registers in which the library routines expect to find their parameters.

FD files also have a number of other uses. C programmers, for instance, can use them to create #pragma statements for generating in-line subroutine calls to external libraries. Incidentally, FD files are not converted to .BMAP files and then used as libraries. The .BMAP file is just a modified function description

file used to provide function details in a form that allows the Basic LIBRARY statement to access and use the real Amiga system libraries!

Q: What is needed to turn C program listings into runnable programs?
A: The C programs found in the pages of magazines represent listings printed from ordinary text files which have been created using programs like ED, MEMACS or some other text editor. These program files are called source code files and are created and saved in just the same way as any other type of text file. To start with then it is necessary to type in the listing to create a C source file.

Unfortunately, several additional steps are needed to convert source files into runnable form. A piece of software called a compiler needs to be used to translate the program lines into code that the Amiga's processor can understand. Next this object code form has to undergo yet another process called linking in order to produce a final runnable program. There's a lot more to the story but most C programming books will outline the steps, and their purposes, in some detail.

Q: How can keypress input be collected from a Shell/CLI type window in a way which prevents echoing and eliminates the need to press the Return key before anything happens?
A: One easy way of collecting keypresses directly if your programs are character stream rather than graphics-based is to open a RAW window and use that for all of the program's I/O operations. RAW: provides unbuffered screen and keyboard I/O and keypresses can be read (unechoed) by a program immediately. Here is a short C example which opens a window and then closes it as soon as a key is touched...

```
#include
#include
main()
{

char c; BPTR file;
file=Open("RAW:20/20/500/100/Test",MODE_NEWFILE);
Read(file, &c, 1); /* Read one char from RAW: */

Close(file); /* Close RAW Window  */
}
```

There is plenty of information about the RAW console device in the *AmigaDOS Manual* published by Bantam Books.

Q: If 68000 assembler programming is as good as everyone says, why do people bother to learn other languages like Basic and C?

A: Because high-level languages like Basic and C are easy to learn and easy to use whereas assembler programming is time consuming and involves a lot of work right from the word go! Learning about the 68000 instructions themselves is easy enough but in order to write 68000 code for the Amiga it is necessary to know about the Amiga's operating system, its library arrangements, and quite a few other things besides. System-wise the Addison Wesley RKM manuals are the official guides to such things and, together with Bantam Book's *AmigaDOS Manual*, provides the most complete description of the system currently available.

The trouble is that these books are not written for beginners – they've been written for users who are, to a large extent, already competent programmers. They do incidentally also assume, among other things, that the reader is C literate and this is one of the main reasons why all Amiga programmers should know something about C.

Most serious Amiga programmers do eventually get around to learning about 68000 assembler but this is not usually in order to give up working with high-level languages.

Such programmers may well write parts of their applications in assembler to gain extra speed, and occasionally write complete programs, but the real benefit of knowing how microprocessors are driven is that you gain an understanding of what computing is all about at the "nuts and bolts" level. A lot of things that often do not make sense to programmers who work solely with high-level languages then suddenly fall into place.

Q: Everyone says that on machines like the Amiga programs need to be properly developed rather than just written while sitting at the keyboard. Surely it's only the end results that matter not the approach used?

A: There's certainly some truth in this argument and if a program is simple enough, and the programmer sufficiently familiar with the problem being coded, there is nothing wrong with creating the program from scratch while sitting at the keyboard. As programs get larger, however, the plain fact is that for most people this approach becomes increasingly more difficult to apply. Put another way – as problems get larger or less familiar the programmer becomes more likely to make mistakes!

Most Amiga programs which do anything useful tend however to be both large and complicated and so most programmers accept that a rather more formal development approach is called for. What is needed is a strategy for identifying more easily handled programming sections and tackling those sections in a systematic manner. It may sound grand but in the main all that is needed is to apply a bit of common-sense to the problem being tackled before

starting to code. Here are a few guidelines...

- Identify the problem and keep a written description of the project to be tackled.

- Break the problem down, using any and every technique that you feel might help. The object of the exercise is to identify smaller, more manageable areas which should, in isolation, be easier to tackle. This is usually an iterative process where you continually ask yourself whether more complex areas may be broken down further. Whenever the answer is "yes" continue the breakdown process. Eventually you get to a point where you will be happy about the coding of various sections of the project.

- Use flowcharts or other design diagrams and notes as a starting point for a basic "program structure". Make lists of the routines that need to be developed, and make a note about those which may be available from library sources.

- Work on, and if possible test, individual routines in isolation. Keep in mind however that they must be compatible with the remainder of the program. Building up programs in this "incremental testing" fashion will make it easier to find bugs because, should any errors occur, they'll usually be connected with the last routine to be added.

- Most importantly, make development notes while you are developing the program, not afterwards. By all means tidy up the notes once the program is complete but don't wait this long before you make any notes at all.

Q: Why on earth do some programmers use subroutines which appear to do nothing at all?

A: You might be forgiven for thinking that subroutines which do nothing serve little purpose but in actual fact such routines can be put to very good use. Take this Basic routine for instance...

```
REM ========================
REM  DO   NOTHING
REM ------------------------
DoNothing: RETURN
REM ========================
```

Supposing you are collecting an input input value X which can have one of six different states and where, depending on the value of X, the program has to execute one of six subroutines. The subsequent code could be based on an

arrangement such as the following...

```
 IF (X>0 AND X<7) THEN ON X GOSUB SUB_A, SUB_B,
SUB_C, SUB_D, SUB_E, SUB_F
```

where SUB_A, SUB_B, SUB_C, SUB_D, SUB_E and SUB_F are the subroutines that perform the associated processing. During development some of these routines might be non-operational (or even non-existent) and so a way is needed of preventing them from being executed. Suppose, for example, that you wanted to prevent subroutine D from being used. One way to achieve this would be to

```
rewrite the IF/THEN conditions using...
IF (X>0 AND X<4) OR (X>4 AND X <7) THEN...
```

but a better idea is just to replace the reference to subroutine D with a reference to a subroutine that does nothing, like this:

```
IF (X<0 AND X>7) THEN ON X GOSUB SUB_A, SUB_B,
SUB_C, DoNothing, SUB_E, SUB_F
```

The approach becomes very useful when large number of test values are involved. One typical example is the execution of routines that are performed when control characters (ie non-printable Ascii characters) are detected. In all probability a program will not wish to act upon all possible control characters and executing a Do Nothing routine for control characters that you do not wish to support provides a nice, clean solution to an otherwise messy verification task.

Q: What is portable code, and how is it written?

A: By definition portable code is code which can be made to run on different machines and in different operating system environments. Having said that, portability is not a rigid concept and in practice there are varying degrees of code portability. Ideally it should be possible to take a program's source code, transfer it to another environment, recompile the source code and find that it still works as expected.

To be honest this level of portability is only possible with programs which have very simple input/output requirements. A more realistic definition is that a program can be considered portable if it can be adapted to work on another system with less effort that it would take to completely re-write it.

One of the areas which can cause difficulties in any language is the I/O stuff – screen handling, file handling, graphics etc. One surefire way to improve the portability of a program is to isolate the code associated with these facilities.

The aim is to eliminate all screen graphics commands, data input or other I/O specific references from the main body of the code and place them in a set of isolated subroutines at the end of the program. To get such a program running on another machine you will probably have to re-write most of those I/O calls but the important point is that you are unlikely to have to alter the main body of the program's code.

Q: Having decided to learn C I've obtained a number of public domain C compilers. Even though programs compile without any errors however they still frequently crash the machine. Why?

A: All it really means when a compiler compiles code without producing error messages is that the code appears to be syntactically correct to the compiler. Doing such things as making library calls without opening the appropriate library, or trying to return memory that hadn't actually been acquired in the first place, can all cause the famous Guru to appear and the solution is undoubtedly to look far more carefully at the code that has been written.

Q: There's a lot of talk about Ansi C library functions but what benefits do they have over ordinary C library functions?

A: The main benefit is that the library contents are tightly standardised, so the routines will be found in the libraries of all Ansi C compilers. This in turn helps make to make it easier to write portable code.

Q: What is ARexx? Is it a replacement for Amiga Basic and if so can real programs be written with ARexx?

A: ARexx (which is pronounced "Ay-Rex") is the Amiga version of the REXX programming language and it is quite easy to learn. Newcomers are going to find their first steps with ARexx much easier than languages such as C and Pascal. It is not a replacement language for Basic but it does have certain things in common with it as far as ease-of-use is concerned. ARexx is a very powerful language and you can most certainly write real programs with it. It is however an interpreted language and, due to the way it does things internally, it can produce slow running programs relative to languages like C.

Q: How does ARexx allow programs to "talk" to each other?

A: What happens is that when ARexx comes across a program statement that it doesn't understand it automatically assumes that it is a command intended for another application. This being so it duly transmits it via a special command interface to the program currently defined by the ARexx current host address

(actually the name of a public message port managed by the host application). All programs that provide ARexx interfaces set up their own, uniquely named, Exec type message ports and it is through these that programs communicate with ARexx.

Q: If I take long-winded C code fragments, such as...

```
if(input_key==QUIT)
{
CloseAllFiles();
DisplayExitMessage(CLOSING);
WaitForReturn();
}
```

and write them along these lines...

```
if(ik==Q){CAF();DEM(C);WFR();}
```

will the compacted form run more quickly than the original code?
A: These tricks frequently improve run-time performance with languages like interpreted Basic but with compiled languages like C they make no difference at all and you should avoid such styles like the plague. Trying to understand compressed code years after it has been written is an absolute nightmare, even for the original author.

Q: What is the best way of avoiding confusion when using pointer variables?
A: This used to be a serious problem and many C programmers chose to adopt a convention of naming pointer variables using a _p suffix so that the variable name itself acted as a reminder that a pointer variable was being dealt with. Nowadays ANSI C compilers can warn against pointer misuse so most misuse slips can be detected at compile time. Nevertheless a fatal error on the Amiga, namely the non-initialisation of pointers, is still commonly found. What happens is that the programmer declares and uses a pointer variable, but fails to check whether it has been set to a proper initial value.

If, for example, this type of code was being used to open an Intuition Screen:

```
screen_p=(struct Screen *)OpenScreen(&NewScreen);
```

The function, in most cases, would execute correctly, ie the OpenScreen() call would set the screen_p pointer variable to the appropriate Screen structure address. Although the code itself is correct however it is not always possible to guarantee that such code will always work because the OpenScreen() function

itself may fail (eg due to lack of available memory). The library call tells you this by returning a NULL (zero) pointer rather than a proper Screen structure address and the result of not checking for this situation is that the program may crash when running on a heavily loaded system. It is because of these types of possibilities that all Amiga system calls should be tested to see that they have been successful.

JARGON BUSTER

compiler – A piece of software which can take a source code file (ie a text file containing the statements which make up a computer program) and convert it into a runnable program consisting of the low-level instructions which the microprocessor understands. The important point about this translation process is that, because it is carried out before the program runs, it only needs to be done once.

crash – An irrecoverable program/system error which causes you to have to reboot (restart) the machine.

echoing – Refers to the automatic display of the appropriate character when a key on a computer keyboard is pressed.

exec – the part of the Amiga operating system that handles things like multitasking and message-based task communications.

Interpreter – A piece of software which does essentially the same job as a compiler but performs the translation to low-level microprocessor instructions in real time, ie each time the program is executed.

I/O – common abbreviation for Input/Output.

library functions – there are actually a variety of uses of the word library as far as the Amiga programmer is concerned but basically all libraries consist of collections of pre-written routines. The C world calls all of these subroutine-type units "functions", hence the routines themselves are called library functions.

pointer – A pointer is a variable which holds a memory address.

RKM – Common abbreviation for the Addison Wesley Amiga ROM Kernel Reference Manuals.

STORAGE DEVICES

Storage devices on the Amiga come in all shapes, sizes and technologies. More and more we have a range of different ways to load and save programs and files, from the humble yet oddly efficient floppy disk to the Write Once Read Many or WORM optical drives. To make some sense of all of this, let's answer some of those frequently asked questions.

Q: There are hundreds of floppy drives to choose from. Which one is the best?
A: They're all going to be pretty much of a muchness. Why? Because the technology which makes the drives possible is simple and reliable, and the one you can afford will probably do the job. If you want more you'll have to read reviews or take personal recommendations from friends or magazines.

Of those we've seen over the last few months, the Evesham Micros External Drive (Evesham Micros) will cost you £47.99. and has all the standard features – slim design, disable switch, through port. The Zappo External Drive (ZCL) £49.99 is once again pretty much the same deal – a slimline drive with sturdy cabling and an on/off switch.

On the more professional side, you might consider the Power Computing Dual Drive (Power Computing). It costs about £125 but is cheap for two drives plus all the little extras. It's not what you might call slimline but it does hold two drive units and has the same key features as the company's PC880B single drive: Blitz Amiga hardware and software which means you can backup disks at lightning speed.

It also has a virus blocking program and neither of the drives click as they have a no-click mechanism. The power overhead problem is solved by the unit having its own power supply. It has a port on the back for daisy chaining and a disable switch so you can turn the drive off when a program doesn't like external drives.

Q: Can I store programs on the PCM Cards which fit in the PCMCIA slot in a 600 and 1200?
A: RAM Cards for the slot have only recently become available, like the New Media Corporation PCMCIA 2Mb card (ZCL). A stunningly tiny package the size of a credit card, and in that you get a full 2-4Mb of extra memory, and it's installed in a second without even touching the rest of your machine. But... it's not a disk but a chunk of RAM. It has no battery-backed memory, so anything you store as a RAM disk will be wiped when you power down. The battery in

the card is merely to power the chips, not to store anything when the machine is off.

Q: How do you fit a hard disk to the Amiga?

A: The A500 and 500 Plus take the Amiga A590 drive, or a host of third-party units like the GVP Impact II and the RocHard. Some of these units are SCSI-based, some are XT IDE-based. The A600 and 1200 can have IDE drives fitted internally, and although at present none are available you'll soon be able to link an external hard drive device of some sort via the PCMCIA slot and suitable interface.

All Amigas with slots, the A2000/1500, A3000 and A4000 can have hard drives mounted in their drive bays or even hard cards which fit in the slots, leaving the drive bays free for 3.5in floppies and a CD ROM drive perhaps.

Q: I have just purchased a 68030 accelerator card. Why doesn't it work with the 2090A card supplied with my 2000 installed?

A: Mostly Amiga peripherals live quite happily together provided certain circumstances prevail. The 2090A is one big exception to this. Most other hard drive or accelerator peripherals can't stand it, and refuse to work when the machine is present in the machine. Manufacturers can't pin down what it is that makes the 2090A so user unfriendly, but it just is. This is possibly the problem. If the accelerator has a hard drive controller on it and some memory then it's possible that it has a jumper set to auto-boot from its own controller rather than the 2090A.

Setting this to off will solve the problem in a small number of cases (GVP cards are pretty good as they seem to able to ignore the 2090A driver). But most barf at the slightest whiff of a 2090A being present. Why do you want to have a 2090A card in your machine anyway? Ah, you have your boot drive connected to it!

That's a tough one. You will have to lose that old 20Mb drive, I'm afraid. The only permanent solution is to get a new SCSI drive controller (if you haven't already got one on your accelerator), and ditch the 2090A down the nearest manhole. That way your accelerator will function beautifully and you'll save an otherwise useless slot.

Q: What is the capacity of a CD ROM disc? How much data is on a CD ROM title?

A: Not many CD ROM discs are full, as the capacity is somewhat large, around 600Mb. CD ROM titles tend to be part data, and part pictures and sound, which goes some way towards bulking out the huge wad of file space in a disc. Sound is often stored as sound samples, but also simply as CD-quality digital music,

the same as a regular CD. Pictures, especially HAM8 pictures, can be as much as 200k, so you can fit a lot of very high quality images for an arcade or adventure game.

Q: Can I save files to my CD ROM Drive? Do I need any extra hardware to do this? Can I have programs transferred to a CD ROM disk for use in my CDTV or A570?

A: The ROM part of CD ROM give you the answer. Read Only Memory is what it says and that what it means (the word "memory" is used slightly confusingly in this context, but substitute the word "storage" and you'll get the picture). Reading a disk is taking stuff off it and loading it into the memory of your computer, and writing is saving things from the computer to the drive. If a drive is read only, then you cannot write to it. The only options which allow you to write to an optical medium are the optical drive and the Write Once Read Many or WORM drive, a similar technology to CD ROM and music CD, but using a different process.

There are companies which will put your data onto a CD for use in your drive, either as a one off, or duplicating a run of 1000s of disks.

Q: I have a CDTV with extra keyboard and disk drive. Is it possible to play a CD in the CD ROM drive and sample it with a sampler plugged into the parallel port?

A: Seems like a good idea doesn't it? Well no it's not possible as the program to allow you to play disks takes over the machine, and you can't really do that and run the sampler software. Maybe this idea will be possible in CDTV version 2, but not at the moment.

Q: What sort of drive is inside my A590 expansion unit? Could I fit an alternative IDE hard drive or even a SCSI drive, as the unit has a SCSI connector on the back?

A: The drive inside a standard A590 unit is an XT IDE drive. This is a very horrible old PC-style specification drive, and SCSI drives are much faster and nicer. You can replace the IDE with a SCSI unit very easily. Firstly the drive you are replacing the internal drive with has to be 3.5in wide or it simply won't fit. Unscrew the casing, and loosen the drive which is in there already. The drive is connected to the board by a ribbon cable into a socket marked "XT DRIVE".

Pull out the plug and discard it and the drive, because you won't be needing them again. Next you need the SCSI drive, and a cable to fit it to the internal SCSI connector (if you're fortunate then you'll have been supplied with a cable

when you bought the drive, if not get one made up. It's a 50-way internal SCSI cable!).

Simply push the connector onto the pins of the socket, and you're away. Ensure that the red edge of the cable is on the same side as the figure "1" on the board. Screw the drive down onto the board and replace the lid. Then you can format and install the drive using your HD programs on your installation disk.

Q: Will any old SCSI drive do for my Amiga? I have access to a number of old SCSI drives from Mac and PC sources, and wonder if I can fit them to my machine?

A: Yes you can fit them. In a 2000 obviously you have the option of installing both 3.5in and 5.25in hard drives in the bays available. On the 500 you can only fit 3.5in internally, although you can plug any size drive externally from the SCSI connector at the back. All you need is the right cabling and a small amendment to your mountlist in the DEVS directory of your system partition. Something like this for an old Seagate ST251N:

```
FH2:  Device = hddisk.device
      Unit   = 3
      Flags  = 0
      Surfaces  = 4
      BlocksPerTrack = 26
      Reserved = 2
      Interleave = 1
      LowCyl = 4 ;  HighCyl = 809
      globvec = -1
      Stacksize 8000
      DosType = 0x444f5301
      FileSystem = L:FastFileSystem
      Buffers = 30
      BufMemType = 0
      Mount = 1
```

SCSI also allows you to add up to six more drives (or laser printers, tape streamers and hard drives) to your system.

Q: I've had a hard disk in my Amiga for a couple of months now, and I'm sure it is a lot slower than it was before. Is this my imagination? If not what can I do about it?

A: The problem is disk fragmentation, a process not unlike memory fragmentation (or middle age brain rot), whereby bits of a program are stored all over the disk as you install and de-install programs and files. As this process

wears on, the drive head has to go further to reach each bit of the file and so it slows down.

The cure is to do one of two things: completely format the disk and restore your files from a backup, as this means every file will be continuous again, or use a de-fragmentation program such as Quarterback Tools, remembering of course to back up your drive with something like Quarterback 5 or AmiBack before you do, just in case something goes horribly wrong. If anything does go wrong, go to option 1 and re-format your drive.

If you have a high capacity drive you may have to spend a lot of time backing it all up, and a lot of floppy disks too. Remember that a floppy only holds about 800k of information, and so parking a 105Mb hard disk on floppies would make a serious dent in 100 disks! The most effective and elegant solution is a SCSI tape streamer device, which puts your data onto a continuous tape and then streams it back when required to capacities up to about 500Mb. Handy if you have two fragged drives, one 300Mb and one 105Mb. Think of the saving in disks alone!

Q: What is drive parking and does it have anything to do with the Park program on my system disk? Do I have to park before I can turn off my drive?
A: The heads on a hard drive are "parked" when they are away from the drive and reset to a ready state, ready that is to go back to the drive surface and start reading data when the drive is turned back on. Most modern SCSI drives are auto-parking, so the Park program isn't needed. Some old XT drives (like the ones in original A590s) are not auto parking however and they need to be manually parked before you turn off the drive. If you don't park the heads after use on a manual parking drive you could risk damage to your drive.

Q: Why can't I daisy-chain more than one or two drives to my Amiga 500?
A: The problem is one of power overheads. The drives (and indeed any peripheral you plug into the Amiga) draw their power through the port, upping the demand through the power supply from the mains supply. The overhead – how much power there is left for other devices when the Amiga has had its share – is very low on the Amiga 500, as the power supply isn't all that highly rated. So if you add more than one drive to the system, you'll have to ensure that it has its own power supply. Oh yes, and of course that means you'll need three plug sockets for your Amiga setup instead of two.

Q: I can't decide whether to buy an A590 hard disk or an A570 CD ROM drive. I'd like to get the A570, but will it have a SCSI port, and the space for 2Mb of memory like the A590?

A: It's an even toss up, but as with most things it's what you want to have a drive for that makes the difference. If you need to have access to CD ROM and CDTV titles, the you have to go for the A570. The A570 doesn't have a SCSI port as standard, but you can fit one as an optional extra. It slots simply and quickly into the back of the unit, and is the same in fact as the one used on the CDTV. That's the simple bit.

The hard bit is that you need an external power supply for your SCSI hard disk, and a container to put it in. If it comes with a case and power supply then all well and good, but if it doesn't you'll have to get one specially made. The power overhead is too shallow to allow something as heavy duty as a hard drive to run as well as the CD ROM drive.

As for RAM, well the A570 comes with enough room for 2Mb of memory.

JARGON BUSTER

daisy chaining – The act of chaining one drive onto the back of another one. Some disk drives have another drive port on the back and you can plug another drive into it. The name comes from the way the drives look when fixed together resembling a daisy chain. Watch out for problems on the A500 though due to the low power overheads.

device name – All drives on an Amiga system have a name, and to make it nice and easy the names are device names, that is to say they have a colon ":" on the end. The internal drive is called DF0:, the next external drive is called DF1:, then DF2: etc. Interestingly enough, if you have a A-Max II Macintosh emulator, the first external drive is the Mac drive, invisible to the system, and so the first Amiga drive is always DF2:! Strange but true.

drive click – Amiga drives click every few seconds or so because they are looking for a disk in the drive. Some PD programs exist to stop this, but some drives actually have a hardware solution to the problem.

expansion slot – A slot in the left-hand cheek plate of an Amiga 500 and 500 Plus, originally designed for the A590 expansion device, the slot has found favour with third-party developers for expansion devices of all shapes and sizes, even big RAM expanders.

IDE – A hard drive interface for the Amiga which allows the connection of XT IDE drives, usually found on PC systems. These drives are usually a good deal slower than their SCSI counterparts.

kilobyte – Usually abbreviated to k. A kilobyte is 1,024 bytes.

megabyte – Also known as Mb or meg. A chunk of 1,024 kilobytes of memory

– in other words a bit more than a floppy disks worth of space.

mountlist – On your boot disk or hard drive partition, the list which describes all your devices and drives which are mounted to the system. The list describes the amount of sectors on the disk and what the drive number is.

PCMCIA – Stands for Personal Computer Memory Card International Association, a group of international computer companies who have created and regulated the developed the PCM card. The credit card slot on the side of the A600/A1200 machines follows this format, and naturally enough the PCM cards which use it.

power overhead – Daisy-chaining drives together is a little bit hazardous on the A500. The power overhead (the power left after some is drawn by each device down the drive port) decreases to very low levels as more drives are added. One or two external drives is all you can really get away with, and it's always safer to add a drive with its own power supply if you can after the first external device, just to make sure.

RAM – An acronym for Random Access Memory. You can read it, you can write to it. You can spend hundreds of quid on it. Wonderful stuff!

ROM – Read Only Memory. Computer chips containing the operating system of the computer which cannot be written to like RAM, only read from.

SCSI – Small Computer Systems Interface, usually pronounced "scuzzy". A system which enables you to connect a number of different types of devices to your computer, like hard disks, CD ROM drives and laser printers.

slimline – Drives used to be big lunking great buggers, and then came the slimline drive mechanism. We said at the time "sometime all drives will be made this way" and we were right, as usual.

VIDEO

Since the launch of the pioneering A1000 back in 1985, one of the Amiga's selling points has been as a computer which can produce graphics for video and TV. The later A1000s even had a built-in colour video output, something which has only recently been re-introduced with the A1200.

Looking back to the teething problems of the A1000, its cost, specification and initial lack of PAL video and graphics software, it is perhaps remarkable

that the Amiga family has survived and prospered so well – and that video is now only one of the myriad uses that today's range of Amiga computers are pressed into service for.

Surprising though it may be, there are many thousands of enthusiasts who are happily working away with almost base-level Amiga 500s and 600s, using them to add titles and other fripperies to their home videos.

Not everyone can afford to upgrade their machines every time the wind changes at Commodore – much as they might like to – and it's important to remember that this core of committed owners has probably done more for the Amiga than the Amiga 2000, 3000 or perhaps even 4000 ever could, regardless of their power.

So lets not forget that Amiga video is by no means confined to high-end accelerated, 24-bit customised, RAM-packed, hard-drive installed super-Amigas. While there's no denying that amazing things can be done with top-end machines and customised kit – and frequently are in the hands of professionals – even the simplest of video applications can be a great challenge to the uninitiated.

So, before your eyes roll back in your head with boredom, how exactly does an Amiga fit in with video production? Quite well actually. There are many facets to video production and the Amiga can be incorporated into many of them.

One of the simplest uses might be to produce a graphic or title which can later be recorded onto videotape; the most complex could be an Amiga fully integrated into a professional video editing suite to supply broadcast quality graphics, animations, titles and more.

Other applications include desktop video, storyboarding, interactive displays, video editing, audio production and 3D animation. Whatever your needs, the Amiga can, in one form or another, cater for them.

Sure, the system won't be as fast or as comprehensive as WaveFront for Silicon Graphics computers or as advanced as a Quantel Paintbox, but it costs only a fraction of the price. The attraction of Amigas is that they are video machines for the people. Or at least they ought to be.

Q: I've noticed that the colour bars provided with some of my video software vary from program to program. Surely, if the idea of colour bars is to assist in the setting up and colour-matching of monitors and computer screens, a standard colour scheme should be adhered to. Why is this not so?

A: Frequently the colour bars provided with some Amiga video and graphics programs are quite wrong – oversaturated and based on a notion of the colours used, rather than the actual hues. While it isn't possible to match the standard colour bars exactly on a standard (non-AA) Amiga it is possible to get

acceptably close.

The following approximation of the standard colour bar results from experiments with an Amiga coupled to a Vectorscope (a piece of video test equipment):

	Red	Green	Blue	Description
White	14	14	14	White
Yellow	10	10	0	Light Mustard
Cyan	0	10	10	Light Greenish-Blue
Green	0	10	0	Brightish Green
Magenta	10	0	10	Purple-Violet
Red	10	0	10	Mid Pillar-Box Red
Blue	0	0	10	Nearly Royal Blue
Black	0	0	0	Black

You can draw up your own chart based on the above settings using a graphics program such as Deluxe Paint. White is the leftmost colour, black the rightmost, and each occupies equal width on the screen. If you use a 352 pixel wide screen then each colour will be 44 pixels wide.

Now, when you set up your Amiga and video monitors you can match them more realistically. If you can't find a photograph of an actual colour bar to refer to use the colour descriptions above to assist you.

Q: I intend producing some 3D animations and I want to record all the frames to video as a single-frame sequence after they have been produced. How can I do this?

A: There are several ways to do this, each of which involves varying amounts of time and money, but all require access to a video recorder which can consistently record to single-frame accuracy. These are generally expensive, so be prepared to spend several thousand pounds (or more) if you plan on buying your own equipment.

You'll also need a single frame controller capable of working with your chosen video deck – such as Optonica's Simpatica or the Nucleus SFC. These are devices which control both the loading of images and the video deck during recording. Simpatica can work with VTRs which have no timecode module, Nucleus SFC requires that the VTR has a timecode reader/generator on board.

It is possible to do such recording work by hand, if you have infinite patience and make precise notes as you go. One technique is to record four frames of the first image, wind back three frames, record four frames of the next image, wind back, etc. Not recommended unless you're a masochist, but it works.

If you plan on working in 24-bit you'll also need a massive storage device –

the bigger the better. A removable hard drive system such as a Syquest would be extremely useful – doubly so if you decide to use an agency such as Alternative Image to collate the frames onto video for you, as all you need then do is carry the Syquest cartridges with you (as long as the agency has its own Syquest hooked up to its Amiga, of course). It is also possible to automatically record each frame as it is generated, but this can put a lot of undue strain on the VTR.

Finally, you'll need a good quality RGB to video converter, such as a well-specified, high-end genlock. Mention this to your bank manager when you see him – I'm sure he'll understand.

Q: How can I transfer my Amiga 500 graphics to videotape?

A: The simplest (and cheapest) way is to get hold of an A520 modulator. Plug this into the Amiga's RGB port and connect the video output to your video deck. Record. That's it.

A more expensive, but more flexible, option is to buy a genlock. They range from under £100 to several thousand pounds for broadcast quality units. As they say, you (usually) gets what you pays for. A genlock will not only let you record graphics and animations to tape but also superimpose (or key) them over live video – for instance from another video deck or video camera.

The cheapest genlocks have little, if any, control over the keying level, though more expensive ones have features such as cross-fading, key inversion (for "keyhole" effects) and selectable key colour. The choice is yours.

Q: I want to add titles and animations to my home videotapes using my Amiga computer but I'm not sure what I need to do this. I've been told I need a genlock. Is this true?

A: Yes, it's true. Not the whole truth though, because there is other equipment, including some vision mixers, which can also do this. You'll also need two video decks – one to play back your source tapes and the other for recording the mixed graphics and video output. And if you plan on adding graphics while you edit, add another video deck and edit controller to the list, as well as a vision mixer.

If you feed the output from the video mixer through the genlock then it becomes possible to overlay graphics while still being able to add effects between the two video sources. This is known as "downstream" keying.

Q: I've seen NewTek's Video Toaster mentioned a lot in American Amiga magazines, and occasionally in British ones, and I'd really like to buy one. But I'm not exactly sure what else I'd need to get the best out of it.

A: As you already know, the Video Toaster is an amazing card which plugs into an Amiga computer and is capable of producing advanced digital video effects, high-quality graphics and 3D animation, among other things.

The first thing you'd need is an Amiga 2000 or 1500 to house the Toaster in. An accelerator would help and a large (upwards of 80Mb) hard drive is almost obligatory. Then you'll need some NTSC format video equipment – cameras, VTRs, time-base correctors and so on, if you want to use the Toaster for video production. At least one NTSC-compatible monitor is also required to display the Toaster's output. Alternatively, you could hook up PAL video equipment via standards converters to achieve the same results.

If you were hoping to work in PAL with the Toaster then forget it. NewTek have been saying they'd have a PAL Video Toaster out "within 18 months" for at least two years, so don't hold your breath. It's a crying shame, but there you go.

An alternative would be to check out the new modules for the OpalVision card – including the Roaster chip – which promise to give the Toaster a run for its money – and in PAL to boot.

Q: Which tape format should I choose for video work?

A: How long is a piece of string? This is one of those classic questions that has several determining factors, particularly those of need and cost.

If you need high quality then you should consider at least Y/C formats – eg S-VHS or Hi-8 – or better, BetaCam SP, MII or D-2 or, alternatively but not quite as good, Hi-Band SP U-matic. All these are far superior to VHS, with BetaCam, MII and D-2 being the best of the bunch.

In general, the higher the quality you record at the better your graphics will look, though this is to some extent dependent on the quality of the RGB-video converter you choose. Recording quality is especially important if you intend editing your material at a later date, as the images will be degraded slightly every time they are copied in the editing process, unless you are fortunate enough to be using digital recording equipment.

Inevitably, the higher the quality the higher the cost of the equipment. Again, it may not be worth investing in expensive kit if you don't plan on using it regularly and you may do better to contact a local company who can hire you the equipment as and when you need it, which should help keep costs down.

Q: Which Amiga is best suited to video work?

A: This depends on what you want to do, how far you intend expanding the Amiga and, of course, how much you wish to spend. For basic work, such as simple titling and graphics, a second-hand Amiga 500 Plus may well fit the bill – especially if it has extra memory and a hard drive included.

For more advanced work there is the new Amiga 1200, though (at the time of writing) software which fully utilises the new graphics modes is still being developed or upgraded. Extra memory and the addition of a hard drive will make this a flexible machine, especially since it has its own video, as well as RGB and RF outputs.

For high-end work the best bet at the moment is perhaps the recently discontinued (and discount priced) Amiga 3000. However, fit a 24-bit card (such as the excellent OpalVision) and you have the basis of a high-quality graphics workstation which can be expanded quite a long way. Everything but the new chipset, in fact.

Finally, the new "serious" Amiga, the A4000, has great potential for video work and especially number-crunching intensive applications such as ray tracing and animation because of its advanced 68040 CPU. Again, adding a 24-bit card will bring out the best, though it might be worth stalling a while to see if Commodore bring out their own full 24-bit machine in the near future – something which shouldn't be ruled out.

Q: What's the point of using a 24-bit card for graphics when a video screen cannot possibly use 16 million colours?

A: Quality, that's what. Because of the 16 million colour palette used in 24-bit graphics it is possible to achieve results which are not possible even with the AA chipset. Smooth gradients and realistic colours contribute to the high-quality appearance even at lower screen resolutions, such as 352 x 290, where only 102,080 pixels (hence 102,080 possible screen colours) are used. In higher resolutions, such as 740 x 580, far more pixels can be in use – providing space for more colours than the AA chips can handle; hence 24-bit will provide a better display altogether.

Q: Do you have any tips for improved video digitising?

A: There are several things which will make for better digitising. If you use a video camera, ensure that it is securely mounted and that adequate light is provided to evenly illuminate the subject. This will avoid the need to either open the iris, increase the video gain or boost the brightness and contrast after grabbing – all of which can add extra noise to the image.

If you digitise from videotape, ensure that your VTR has a solid freeze-frame – this will help avoid jitters or interference. Use the best quality tape possible.

In both cases S-VHS or Video8 generally provide better results than composite video (though this does depend on the make of digitiser), so use one with Y/C feeds if possible.

Q: Which graphics modes should I use for video work?

A: Use the highest screen resolution you possibly can, though you may have to sacrifice colours to get there, depending upon which Amiga you are using. At the very least you ought to be using an Interlaced mode. You should also use an overscan size if you wish your images to reach right to the edges of the video screen – for example Deluxe Paint's 352 x 580 Interlaced overscan size. There's nothing more disconcerting than a smoothly animated logo popping up part way across the screen.

Q: Is there any particular kind of monitor which I should use with my Amiga for video work or will any one do?

A: For the most flexibility a monitor with both RGB and composite input will serve best for video work. It will give you the choice of connecting directly to the Amiga, or from a genlock with RGB pass-through, and also let you view video signals directly.

The RGB signal from the Amiga is superior to either RF or composite video and this should be your choice whenever possible.

I would recommend using a standard TV set only as a last resort, as the RF signal output from the Amiga (or modulator) will make some operations, such as designing fine graphics, quite difficult.

Although you might be tempted to buy a multisync monitor so that you can have the benefits of flicker-free viewing, you should seriously consider buying one which will sync down to 15.6kHz – the frequency of the normal Amiga output. This is because animations and other moving graphics will not display normally on many multisyncs – hence the need to switch over from time to time.

Q: Is it OK to save my 24-bit images as JPEG files in order to reduce storage space?

A: Yes and no. If the images are part of an animation sequence it would be better to keep them in uncompressed format until you have produced the finished version of the animation and you are happy with it. This is because the JPEG compression method works only on single images and cannot compare the current frame with its neighbours, which could result in a "shimmering" effect after the frames are decompressed and animated, even if you use the best possible compression ratio.

Additionally, JPEG causes slight quality loss which, although all but imperceptible at the highest quality, could still be seen in an animation. The best method to use would be MPEG compression, but this isn't yet practical on the Amiga.

Q: Some of the colours in my graphics seem to be rather smeary when I transfer them to video. Is there anything I can do about this?

A: This is a problem with all video systems, but which decreases the further up the quality ladder you get. Systems like VHS come off worst, as their recording bandwidth is relatively poor and the signals have to be coded and decoded to and from composite for recording and replay. Component systems fare better and broadcast formats just about have it cracked.

If you use VHS or even S-VHS or Video8 you should consider replacing the offending colours – which are commonly bright reds and blues – with less saturated tones instead. The visual result will be only slightly different but the smeariness could be substantially reduced. Using a better quality genlock or video encoder can also help

JARGON BUSTER

24-bit – A high quality graphics format with a palette of over 16 million colours, made possible by using 24 bitplanes simultaneously.

anti-aliasing – A technique for smoothing out the jaggies created by pixels (which are the building blocks of computer graphics) by adding intermediate toning between adjacent colours.

bitmap font – A set of special pixel images arranged in the shape of letters. With the exception of Compugraphic and other vector-type fonts, all of the Amiga's fonts are bitmapped.

black burst – black and burst. A generated video signal without the actual picture information which can be used for providing synchronisation information to video equipment or for "blacking" new videotapes prior to editing.

BNC – British Naval Connector. A type of video connector which has a secure, bayonet-style fitting.

character generator – A video device used to generate text for transfer to video. There are several character generator programs for the Amiga.

chroma key – One video image can be superimposed over another by replacing selected areas of the superimposed image according to its colour (Chroma) level. Key levels are usually adjustable and blue is the most frequently used key colour, since humans (the most frequently keyed subject) contain very little blue in their skin tones.

chrominance – The part of a video signal which contains the colour information.

component video – A video signal where components such as chrominance and luminance are processed separately from each other to achieve a better quality picture. An example of is the Y/C format used in S-VHS and Hi-8.

composite video – or CVBS or FBAS. A video signal which contains its chrominance and luminance information in one combined signal.

contrtol track – A video track onto which regular pulses are recorded so that the position of the tape can be read by a VCR as time elapsed. Unlike timecode, index points must be set in order to locate other points on the tape.

digital effects – Video effects which change the size and/or position of a mixed video signal. Requires the use of Framestores.

digitising – The conversion of data originating in the real world (analogue) into a form which a computer can understand (digital). Could be sound or pictures, for example.

EBU – European Broadcasting Union – the European equivalent of SMPTE.

framestore – Digital storage capable of holding a complete frame of video. Used in standards convertors and digital effects generators.

genlocking – A way of synchronising one video source (eg Amiga) with another (eg videotape) to enable stable wipes, mixes and other effects, including keying, between the two sources.

HAM – Hold and Modify. An Amiga display mode allowing 4,096 colours to be displayed at once, though with certain restrictions.

HAM8 – Hold and Modify 8. A graphics mode available with the Amiga's new AA chipset to provide displays of up to 262,144 colours for very impressive-looking graphics.

IFF – Interchange File Format. A set of standard file formats which allow data from different graphics, sound or animation programs (for example) to be used by compatible software.

luma key – Superimposition of one image over another by replacing selected areas of the superimposed image according to their brightness.

luminance – The mono part of a video signal which carries the Luma or brightness information.

MIX – aka dissolve. A process where one image is simultaneously faded up over another which is fading down.

morphing – Very popular at the moment, computer-assisted video/graphics metamorphosis as seen in Terminator 2, TV commercials and much more!

NTSC – National Television Standards Committee. The TV colour coding system used in the USA and other countries, having 525 lines, 60 fields and 30 frames/second. It is often, and perhaps unfairly, japed at as Never Twice the Same Colour by PAL users.

PAL – The world's other main TV colour coding system (with the exception of the French SECAM system). PAL refers to Phase Alteration Line. It has 625 lines at 50 fields and 25 frames/second.

pixel – Picture Element. The smallest unit of display on a computer screen.

RGB – Red, Green and Blue colour components of a video signal.

SEG – Special Effects Generator, aka vision mixer or switcher. A piece of video equipment for producing video transitions and effects, eg wipes, dissolves and keys.

SMPTE – Society of Motion Picture and Television Engineers. A professional organisation in the USA which sets technical standards for American broadcasting. Hence SMPTE timecode.

SPG – Sync Pulse Generator. A device which produces sync pulses for video and can be used to lock all connected pieces of equipment together so that they run in full synchronisation.

time base corrector – For vision mixing and other effects from videotape the video signals have to be in exact synchronisation. Feeding each signal through a time base corrector causes them be re-timed and brought into the correct sync for clean mixing to occur.

timecode – A numerical coding system recorded onto audio or videotape to uniquely identify hours, minutes, seconds and frames etc to enable accurate location of any point on the tape. There are various timecodes in use, depending on the type of equipment. These include VITC (Vertical Interval Timecode), LTC (Longitudinal Timecode), RCTC (Re-writable Consumer Timecode) and MTC (Midi Timecode).

vector font – A font produced from mathematical formulae, such as Agfa's Compugraphic fonts, which allows any letter to be smoothly scaled – with none of the jaggies which plague bitmap fonts.

VTR – Video Tape Recorder. Used to refer to open-reel video machines but now often used interchangeably with VCR.

VCR – Video Cassette Recorder.

wipe – A visual transition between two images, where the edge of one progressively obscures or reveals the other.

Have you ever cursed your keyboard or detested your disk drive? Ever thought confusion might be a terminal affliction? **Amiga Computing** *helps solve the trickier Amiga problems every month, and as an aid to all our readers, new and old, we've decided to collect together some of the most frequently asked questions and have our panel of experts answer them.*

Topics covered range across the entire spectrum of Amiga misery and should ensure that for those more common ailments there is always a cure ready to hand.

Programming ● Monitors
Disk drives ● Graphics
Printers ● AmigaDOS
Video ● Sound

All examined and explained!

AMIGA COMPUTING

europress ENTERPRISE